

# A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints

H. Lantéri\*, M. Roche, O. Cuevas, C. Aime

*Faculté des Sciences, UMR 6525 "Astrophysique", Université de Nice, Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 2, France*

Received 21 February 2000; received in revised form 27 November 2000

## Abstract

The aim of the present paper is to give a general method allowing us to devise maximum-likelihood multiplicative algorithms for inverse problems, and particularly for signal and image restoration with non-negativity constraint. We consider the case of a Gaussian additive noise and that of a Poisson process. The method is founded on the Kuhn–Tucker first-order optimality conditions and the algorithms are developed to satisfy these conditions. The proposed method can be used for any convex function whose definition range includes the domain of constraints. It allows to obtain generalized forms of classical algorithms (ISRA and RLA) and to unify the method for obtaining these algorithms. We give relaxed forms of the algorithms to increase the convergence speed; moreover, the effect of the constraints is clearly shown. For a better understanding of the method to take into account the constraints, we express the non-negativity constraint using different functions and we reach a large class of algorithms that can be analyzed as descent algorithms. Then, we can justify and analyze the behavior of several algorithms suggested in the literature. The particular displacement directions appearing in such algorithms are evidenced and the convergence speed is analyzed. The algorithms are applied for simulated data, to a two-dimensional deconvolution problem, to show their performance and effectiveness. A support constraint is taken into account implicitly in the algorithms. Our method can be extended to more general hard constraints on the extreme values or on the support of the solution and a regularization of the problem can be easily introduced in the method. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Inverse problems; Deconvolution; Signal restoration; Maximum likelihood; Constrained optimization

## 1. Introduction

The image restoration problem consists in the reconstruction of the best estimate of an object “ $x$ ” from the knowledge of a blurred image “ $y$ ” con-

taminated by noise. In the case of a general inverse problem, the transformation suffered by “ $x$ ” is described by a Fredholm integral equation of the first kind:

$$\tilde{y}(r) = \int h(r, \alpha)x(\alpha) d\alpha, \quad (1.1)$$

where  $\tilde{y}(r)$  is the noiseless blurred signal. Here, the kernel  $h(r, \alpha)$  will be assumed positive. If the kernel

\* Corresponding author.

*E-mail address:* hlanteri@unice.fr (H. Lantéri).

$h$  is space invariant, (1.1) becomes

$$\tilde{y}(r) = \int h(r - \alpha)x(\alpha) d\alpha, \quad (1.2)$$

i.e. a convolution equation for which “ $h$ ” is referred to as the point spread function (PSF).

The algorithmic methods and results presented here are developed within the general inverse problem framework described by (1.1), whereas the deconvolution problem (1.2) is addressed to check the algorithms.

In any case, we seek to estimate “ $x$ ” from a noisy version “ $y$ ” of  $\tilde{y}$ . This is an ill-posed problem in the sense of Hadamar, and the difficulties specific to such problems have been extensively described in the literature [2–4,11]. A prior knowledge of the properties of the admissible solutions is required to obtain stable and physically suitable results. For the image deconvolution problem, the classical constraint concerns the non-negativity of the solution; such constraint has been frequently analyzed in the literature [31,35,37–39].

A general method that can be mentioned is the projection onto convex sets (POCS). This well-known approach (see for example [43,8] and references therein) was applied to the Poisson noise case by Stark et al. [36]. However, it gives rise to some difficulties in the definition of the projections connected to the likelihood, and the corresponding algorithms are not multiplicative.

Beside this method, most of the signal restoration iterative multiplicative algorithms proposed in the literature are founded on an explicit likelihood maximization [23,44]. Our method is developed in this context for two classical processes, the Poisson process and the additive zero mean, white Gaussian noise. The main objectives of our paper are to propose a general method to devise the algorithms for likelihood maximization under constraint of non-negative solutions and to show how to express these algorithms in a multiplicative form [17].

We focused on the multiplicative algorithms because they take a very simple form clearly ensuring the non-negativity of the solution, and because a support constraint on the solution is implicitly taken into account. Moreover, their convergence does not require specific procedures. On the other hand, in the classical form of the algorithms, the

convergence speed is fixed, and an acceleration method can be useful. From a general point of view, their obvious simplicity and the absence of a general method to devise them, make difficult an analysis of their behavior and a regularization of the problem.

The approaches already published are basically different in the two cases: for the Poisson process, the Expectation-Maximization (EM) technique [12,20] is always mentioned. It leads to the Richardson–Lucy algorithm [28,32], well known in the fields of astrophysics and medical imaging. For the Gaussian noise, Daube-Witterspoon and Muehlenner [10] proposed on an intuitive basis, the Image Space Reconstruction Algorithm (ISRA), extending the works of Chahine [7] and Gold [18].

For both the noise processes, we are faced to the problem of minimization of a functional  $J(x)$  depending on the particular noise process, with a constraint for non-negative solutions.

In the Gaussian case, a frequently used technique is based on the projected iterative descent method (see for example [21,22] and references therein). This method gives good results because the definition range of corresponding functional is unbounded. Consequently, in the first step of the method (unconstrained minimization), the solution “ $x$ ” can be searched for out of the domain of the constraints, then projected on this domain. Evidently, multiplicative algorithms cannot be obtained by such a method.

On the contrary, for the Poisson process, we emphasize the fact that this procedure cannot be used. Indeed, the definition range of the objective function used in that case is bounded. With the procedure previously defined, there is a risk, during the step of unconstrained minimization, to go outside the domain of definition of the objective function. In this case evidently, the method fails.

Fortunately, in all the cases considered here, the domain of the constraints is completely included in the definition range of the objective function, then if the solution is searched for inside the constraints range, the objective function remains always defined.

Founding on these considerations, we propose to devise the algorithms following the general rule: at each step of the iterative method, the constraints

must be taken into account first. Only after that, the descent step can be performed to compute the best estimate in the space of constrained solutions.

Although the solution space is reduced by the constraints, the problem is ill-posed, and a regularization by a smoothness constraint remains necessary. In the present paper, we do not consider such a problem that will be treated elsewhere. We use simulated data, and the optimal iteration number is merely determined by comparison with the true solution.

For the sake of clarity, we use the matrix notation corresponding to the discrete problem. Eqs. (1.1) and (1.2) are then written in the form of the linear system  $\tilde{y} = Hx$ ; the matrix  $H$  has positive terms and no supplementary assumption is made to preserve the generality of the results. Indeed, although numerical illustrations are given for a deconvolution problem, that is in the space invariant case, the analysis in matrix form obviously holds for general space variant inverse problems.

This paper is organized as follows. In Section 2, after a brief recall of the basic principle of constrained minimization, we develop the general method to obtain iterative multiplicative algorithms for minimization of a convex functional  $J(x)$  with non-negativity constraints. The method is applied to the Poisson process in Section 3 and to the Gaussian noise process in Section 4. In this section, we propose an algorithm for which the relaxation plays a major role. Section 5 is devoted to a few remarks on the properties of these algorithms. In Section 6, the particular features connected to the non-negativity constraint are evidenced, and we show that faster multiplicative algorithms can be derived if the same constraints are expressed differently. Illustrations are given in Section 7 for a two-dimensional image deconvolution problem, and the speed-up effects due to the relaxation are analyzed for different types of images. Conclusions are given in Section 8.

In Appendix A, we develop a general method for obtaining the accelerated algorithms given in Section 6, while Appendix B is devoted to a formal comparison with Rosen's [1] projected gradient method. We show that the proposed algorithms can be considered as a weighted version of this method, or more generally as a projected descent

direction method. The different forms of the algorithms are summarized in Appendix C.

## 2. The general algorithmic method

The general problem can be stated as

$J(x)$  is a convex function  
 Minimize  $x: J(x)$   
 with the constraint  $x \geq 0$ ,

$x^*$  is a solution of this problem if and only if the Kuhn–Tucker first-order optimality conditions [1,29] are verified at  $x^*$ .

We propose to devise algorithms based on these conditions. Let  $L(x, \lambda)$  the Lagrange function associated to this problem, in the case of constraint for non-negative solutions, this function is expressed as

$$L(x, \lambda) = J(x) - (\lambda, g(x)), \tag{2.1}$$

where  $\lambda$  is the Lagrange multipliers vector, with components  $\lambda_i \geq 0 \forall i$ ;  $(\lambda, g(x))$  represents the inner product;  $g(x)$  is a function expressing the constraints, it must be increasing and positive when the constraints are inactive ( $x > 0$ ) and zero for active constraints ( $x = 0$ ), moreover the zeros of  $g_i(x^*)/[Vg(x^*)]_i$  must be the same than those of  $g_i(x^*)$ .

The Kuhn–Tucker conditions at the optimum  $x^*, \lambda^*$  are [1,29]:

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= 0 \\ \Leftrightarrow \lambda_i^* [Vg(x^*)]_i &= [VJ(x^*)]_i \\ \Leftrightarrow \lambda_i^* &= \frac{[VJ(x^*)]_i}{[Vg(x^*)]_i} \quad \forall i, \end{aligned} \tag{2.2a}$$

$$g(x^*) \geq 0 \Leftrightarrow g_i(x^*) \geq 0 \quad \forall i, \tag{2.2b}$$

$$\lambda_i^* \geq 0 \Leftrightarrow \lambda_i^* \geq 0 \quad \forall i, \tag{2.2c}$$

$$\lambda_i^* g_i(x^*) = 0 \Leftrightarrow \frac{[VJ(x^*)]_i}{[Vg(x^*)]_i} g_i(x^*) = 0 \quad \forall i. \tag{2.2d}$$

Taking into account the properties of  $g(x)$ , this last equation reduces to

$$[VJ(x^*)]_i g_i(x^*) = 0 \quad \forall i. \tag{2.2e}$$

Conditions (2.2d) and (2.2e) represent, in fact, two complementary conditions:

$$\lambda_i^* = 0 \equiv \frac{[\nabla J(x^*)]_i}{[\nabla g(x^*)]_i} = 0 \quad \text{if } g_i(x^*) > 0$$

$$\Leftrightarrow \text{Inactive constraint,} \tag{2.3a}$$

$$\lambda_i^* > 0 \equiv \frac{[\nabla J(x^*)]_i}{[\nabla g(x^*)]_i} > 0 \quad \text{if } g_i(x^*) = 0$$

$$\Leftrightarrow \text{Active constraint.} \tag{2.3b}$$

The function  $g(x)$  used to express the constraints is of fundamental importance for the final form of the algorithm. In the simplest case, we can choose  $g(x) = x \Leftrightarrow g_i(x) = x_i \quad \forall i$ .

To solve (2.2d) and (2.2e), we take into account the fact that  $-\nabla J(x)$  is a descent direction for the unconstrained problem [29], and we use the successive substitutions method [41] to write the algorithm in the general gradient form

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} f_i(x^{(k)}) x_i^{(k)} [-\nabla J(x)]_i, \tag{2.4}$$

where  $\alpha_i^{(k)} > 0$  is a relaxation factor, and  $f_i(x^{(k)})$  is a function having positive values when  $x^{(k)}$  satisfies the constraints; this function will depend on the form of  $J(x)$  and  $g(x)$ , and of the particular properties wanted in the algorithm, mainly a multiplicative form. The function  $f_i(x^{(k)})$  will be explicitly written for each case developed in this paper.

Algorithm (2.4) can be written in such a form only because the constraints are in a simple linear form that allows us to obtain an explicit expression of the Lagrange multipliers (2.2a). Relation (2.4) is the basic form we shall use when the function  $g(x)$  will be changed to express the same constraints.

The relaxation of the algorithm allows a control of the convergence speed and gives a simple solution to the problem of acceleration of the algorithms frequently mentioned in the literature [5,19,30].

We now analyze the convergence properties of (2.4) toward a solution for which all the KT conditions (2.2b) and (2.2c) are verified.

Condition (2.2b) imposes restrictions on the stepsize; indeed to retain the generality of the argumentation we consider that  $J(x)$  is not necessarily defined for all the values of “ $x$ ”, but that it is always

defined in the domain of the constraints. Therefore, condition (2.2b) must be imposed first; this condition is fulfilled if at each iteration, we have  $x_i^{(k+1)} \geq 0 \quad \forall i$ , when  $x_i^{(k)} \geq 0 \quad \forall i$ , that is, from (2.4):

$$1 + \alpha_i^{(k)} f_i(x^{(k)}) [-\nabla J(x^{(k)})]_i \geq 0 \quad \forall i. \tag{2.5}$$

For  $[\nabla J(x^{(k)})]_i < 0$  condition (2.5) is always satisfied. The non-negativity does not introduce any restriction on the stepsize.

For  $[\nabla J(x^{(k)})]_i > 0$  we must have for each corresponding index “ $i$ ”:

$$\alpha_i^{(k)} \leq \frac{1}{f_i(x^{(k)}) [-\nabla J(x^{(k)})]_i}. \tag{2.6}$$

We have then a set of the maximal values of the stepsize ensuring the fulfillment of the constraints for each component separately.

The maximum stepsize  $\alpha_P^{(k)}$ , independent of “ $i$ ”, for a non-negative solution will be given by

$$\alpha_P^{(k)} = \text{Min}_i \frac{1}{f_i(x^{(k)}) [-\nabla J(x^{(k)})]_i} \quad \forall i \tag{2.7}$$

such that  $[\nabla J(x^{(k)})]_i > 0$  and  $x_i > 0$ .

Concerning the convergence, the optimal stepsize  $\alpha_C^{(k)}$  independent of “ $i$ ”, must be computed by a line search procedure [1], in the interval  $]0, \alpha_P^{(k)}]$ , in the direction:

$$d^{(k)} = \text{diag}[f_i(x^{(k)})] \text{diag}[x_i^{(k)}] [-\nabla J(x^{(k)})]. \tag{2.8}$$

This direction is no longer the negative gradient but it remains a descent direction for  $J(x)$ . The general algorithm can then be written in the form

$$x^{(k+1)} = x^{(k)} + \alpha_C^{(k)} d^{(k)}. \tag{2.9}$$

Proceeding in such a way we are ensured that the algorithm converges and that the solution is never searched for out of the definition range of  $J(x)$ .

Finally, the KT conditions (2.3) are satisfied since:

- if the solution  $x_i^* > 0$  then, from (2.4), clearly  $\nabla J(x^*) = \lambda^* \equiv 0$ ,
- if  $x_i^* = 0$  and  $[\nabla J(x^*)]_i < 0$  we arrive to a contradiction because  $\{1 + \alpha_i^{(k)} f_i(x^{(k)}) [-\nabla J(x^{(k)})]_i\}$  will be greater than “1” in the neighborhood of  $x^*$  and we can never reach the solution.

The procedure described above constitutes the basis of the method we develop in the following sections for the two basic cases: the Poisson process and the Gaussian additive noise.

### 3. Poisson noise process

For a perfect photon detection, the intensity in the pixel “ $i$ ” is a random variable that follows a Poisson law of mean  $(Hx)_i$ . The likelihood can be written as [27]

$$L(x) = P(y/x) = \prod_i \frac{[(Hx)_i]^{y_i}}{y_i!} \exp[-(Hx)_i], \quad (3.1)$$

where  $(Hx)_i = \sum_j h_{ij}x_j$ .

Using Stirling’s formula, the negative Log-likelihood becomes

$$T(x) = -\text{Log}[L(x)] \approx \sum_i [(Hx)_i - y_i] + y_i \text{Log} \frac{y_i}{(Hx)_i}. \quad (3.2)$$

This function is known as the Csiszär [9] I-divergence measure between  $(Hx)$  and  $y$ . This measure generalizes the Kullback divergence or cross-entropy measure to accommodate functions whose integrals are not constant, as they would be if they were probability distributions [6]. Dropping in  $T(x)$  the terms independent of “ $x$ ”, the problem can be formulated as

$$\text{Minimize}_x: D(x) = \sum_i (Hx)_i - y_i \text{Log}(Hx)_i$$

with the constraint

$$x_i \geq 0 \quad \forall i. \quad (3.3)$$

The functional  $D(x)$  is defined only if  $(Hx)_i > 0 \quad \forall i$ . Clearly, because the elements of  $H$  are positive,  $D(x)$  is always defined if  $x > 0$ , this is why, in the iterative method proposed, the non-negativity condition will be imposed first. Only then, we will compute the best estimate of the solution.

We can easily obtain

$$\nabla D(x) = H^T \text{diag}[1/(Hx)_i](Hx - y).$$

Following the method developed in Section 2, with  $x_i \geq 0 \quad \forall i$ ,  $\sum_j h_{j,i} = a_i > 0$ , and  $f_i(x) = 1/a_i$ , we obtain from (2.4) the relaxed iterative algorithm:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\alpha_i^{(k)}}{a_i} x_i^{(k)} [H^T \text{diag}[1/(Hx^{(k)})_i] \times (y - Hx^{(k)})]_i. \quad (3.4)$$

Denoting for sake of clarity:  $y/Hx = \text{diag}[1/(Hx)_i]y$ , we can also write

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\alpha_i^{(k)}}{a_i} x_i^{(k)} \left[ H^T \left( \frac{y}{Hx^{(k)}} - 1 \right) \right]_i. \quad (3.5)$$

Following (2.7),  $\alpha_p^{(k)}$  will be given by [30]

$$\alpha_p^{(k)} = \text{Min}/i \frac{a_i}{a_i - [H^T(y/Hx^{(k)})]_i} \quad \forall i, \quad \text{such that } a_i - [H^T(y/Hx^{(k)})]_i > 0 \quad (3.6)$$

and  $x_i^{(k)} > 0$ .

Clearly  $\alpha_p^{(k)} \geq 1 \quad \forall k$ , the exact value  $\alpha_p^{(k)}$  must be computed at each iteration step. The value of  $\alpha_c^{(k)}$  must be computed by a line search procedure in the range  $]0, \alpha_p^{(k)}]$ , in the descent direction

$$d^{(k)} = \text{diag} \left[ \frac{x_i^{(k)}}{a_i} \right] \left[ H^T \left( \frac{y}{Hx^{(k)}} - 1 \right) \right]. \quad (3.7)$$

Then, the algorithm can be written in the general form (2.9).

In the particular case  $\alpha_c^{(k)} = 1 \quad \forall k$ , the negative term exactly cancels and algorithm (3.5) can be written in a pure multiplicative form

$$x^{(k+1)} = \text{diag} \left[ \frac{x_i^{(k)}}{a_i} \right] \left\{ H^T \text{diag} \left[ \frac{1}{(Hx^{(k)})_i} \right] y \right\} \quad (3.8)$$

or alternatively, in the form

$$x^{(k+1)} = x^{(k)} + \text{diag} \left[ \frac{x_i^{(k)}}{a_i} \right] \times \left\{ H^T \text{diag} \left[ \frac{1}{(Hx^{(k)})_i} \right] (y - Hx^{(k)}) \right\}. \quad (3.9)$$

Moreover, in the deconvolution problems the matrix  $H$  is such that  $a_i = 1 \quad \forall i$ . It was shown in [34] that with  $\alpha_c^{(k)} = 1 \quad \forall k$ , the algorithm converges. Algorithm (3.8) was proposed in astronomy by

Richardson [32] and Lucy [28], or as the EM method [12,34] in the field of Medical Imaging. Obviously, all the successive estimates remain positive if the initial estimate is positive. Moreover, if a component of the solution becomes equal to zero, it remains equal to zero for all successive iterations.

The forms (3.4) or (3.9) allow an analysis of the behavior of the algorithms in the context of the descent methods. The components of the direction of the displacement are the components of the negative gradient weighted by the factors  $\beta_i = x_i^{(k)}/a_i$  remains always positive or zero, so the descent property of the algorithm is retained. The proportionality of the weights  $\beta_i$  with  $x_i^{(k)}$  is typical of the positivity constraint and can be more clearly understood if  $x_i^{(k)}$  is considered as  $(x_i^{(k)} - 0)$ , the distance between the current estimate and the constraint.

**4. Gaussian noise**

In this case, following for example Llacer and Nuñez [27] or Katsaggelos [21], we consider that the data “ $y_i$ ” are independent random variables, corrupted by an additive, zero-mean, white Gaussian noise “ $n$ ”, with variance  $\sigma_i$ . The image formation model may be written as  $y = Hx + n$ . For a given pixel “ $i$ ”, the data “ $y_i$ ” is the sum of a deterministic part  $(Hx)_i$ , which is the mean of the intensity in the considered pixel, and of a random noise component “ $n_i$ ”. The likelihood, that is the conditional probability to have data “ $y$ ” knowing the mean “ $Hx$ ” is given by

$$L(x) = P(y/x) = \prod_i \frac{1}{\sigma_i \sqrt{2\pi}} \exp[-(y_i - (Hx)_i)^2 / 2\sigma_i^2]. \quad (4.1)$$

The corresponding Log-likelihood is then

$$\text{Log}[P(y/x)] = C - \frac{1}{2} \sum_i (y_i - (Hx)_i)^2 / \sigma_i^2, \quad (4.2)$$

dropping the additive constant  $C$ , the maximization of the likelihood is equivalent to the minimization of

$$G(x) = \frac{1}{2} \sum_i [y_i - (Hx)_i]^2 / \sigma_i^2 = \frac{1}{2} \|y - Hx\|_R^2, \quad (4.3)$$

where  $\| \cdot \|$  represent the Euclidean norm and where  $R$  is the weighting diagonal matrix:

$$R = \text{diag} \left[ \frac{1}{\sigma_i^2} \right]. \quad (4.4)$$

Such functional  $G(x)$  is defined for all values of “ $x$ ”, then all the methods indicated in Section 1 [22] can be used to impose the non-negativity constraint; however, only the method proposed here allows to reach the multiplicative form of the algorithms.

We have

$$\nabla G(x) = H^T R H x^{(k)} - H^T R y. \quad (4.5)$$

Following the method developed in Section 2, we use in the simplest case  $f_i(x^{(k)}) = 1 \forall i, k$ , and then the algorithm deduced from (2.4) writes

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} x_i^{(k)} (H^T R y - H^T R H x^{(k)})_i. \quad (4.6)$$

In this form, we observe that the components of the displacement are the components of the negative gradient, weighted by the positive factors:  $\beta_i^{(k)} = x_i^{(k)}$ .

From (2.7), we have

$$\alpha_p^{(k)} = \text{Min}_i / \frac{1}{(H^T R H x^{(k)} - H^T R y)_i} \forall i$$

such that  $x_i^{(k)} > 0$  (4.7)

and  $[\nabla G(x^{(k)})]_i > 0$ .

The optimal stepsize  $\alpha_C^{(k)}$  is computed by line search in the range  $]0, \alpha_p^{(k)}]$  in the direction

$$d^{(k)} = \text{diag}[x_i^{(k)}] (H^T R y - H^T R H x^{(k)}). \quad (4.8)$$

The algorithm can be written in the general form (2.9).

In our problem, the matrix  $H$  and  $R$  have positive entries, and all the components of the initial estimate have positive values to satisfy the constraint, then we have  $[H^T R H x^{(k)}]_i \geq 0 \forall i, k$ . With the aim to obtain multiplicative algorithms, we use for  $f_i(x^{(k)})$  the particular function

$$f_i(x^{(k)}) = \frac{1}{[H^T R H x^{(k)}]_i}. \quad (4.9)$$

The relaxed algorithm then becomes from (2.4):

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} \frac{x_i^{(k)}}{[H^T R H x^{(k)}]_i} \times (H^T R y - H^T R H x^{(k)})_i. \quad (4.10)$$

The components of the displacement are now, the components of the negative gradient, weighted by the positive factors

$$\beta_i^{(k)} = \frac{x_i^{(k)}}{(H^T R H x^{(k)})_i}. \quad (4.11)$$

We can also write for later use

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} x_i^{(k)} \left[ \frac{(H^T R y)_i}{(H^T R H x^{(k)})_i} - 1 \right]. \quad (4.12)$$

If the relaxation factor is computed so that the successive estimates remain positive, we will have  $\beta_i^{(k)} \geq 0 \forall i, k$ , and the descent properties of the algorithm are preserved.

The particular choice of  $f_i(x^{(k)})$  can now be clearly understood: it allows us to obtain a form similar to (3.5). However, the form (4.12) allows a simple analysis of the behavior of the algorithm concerning the non-negativity of the solution.

In the general case, from (2.7),  $\alpha_P^{(k)}$  is computed as

$$\alpha_P^{(k)} = \text{Min}/i \left[ \frac{1}{1 - [(H^T R y)_i / (H^T R H x^{(k)})_i]} \right] \forall i$$

such that  $[(H^T R H x^{(k)} - H^T R y)_i > 0]_i > 0$  (4.13)

and  $x_i^{(k)} > 0$ .

This value is always higher or equal to 1, therefore, an acceleration of the algorithm can be expected if the optimal value of the stepsize  $\alpha_C^{(k)}$  is computed by line search in the range  $]0, \alpha_P^{(k)}]$ , in the direction

$$d^{(k)} = \text{diag} \left[ \frac{x_i^{(k)}}{[H^T R H x^{(k)}]_i} \right] (H^T R y - H^T R H x^{(k)}). \quad (4.14)$$

The algorithm can be written in general form (2.9).

In the particular case  $\alpha_C^{(k)} = 1 \forall k$ , the negative term in (4.12) is exactly canceled and we obtain  $\forall i$

$$x_i^{(k+1)} = x_i^{(k)} \frac{[H^T R y]_i}{[H^T R H x^{(k)}]_i}. \quad (4.15)$$

This purely multiplicative algorithm called Image Space Reconstruction Algorithm (ISRA) was

initially proposed in the literature by Daube-Witherspoon and Muehllehner [10]. Its convergence was analyzed later by De Pierro [13–15] and Titterton [40].

### 5. Remarks about these algorithms

(1) The algorithms in the forms (3.4) and (4.10) for Poisson and Gaussian noise, respectively, show the similar roles played by the matrix  $\text{diag}(1/(Hx^{(k)})_i)$  and by the weighting matrix  $R = \text{diag}(1/\sigma_i^2)$ . These quantities both correspond to variances of the signal. For the Gaussian process,  $\sigma_i^2$  is the variance of the intensity in a pixel. For the Poisson process, the mean  $(Hx^{(k)})_i$  also represents the variance of the signal.

(2) In the Gaussian case, the minimization without constraints of the quadratic form  $G(x)$  leads to linear algorithms, thus the non-linearity in algorithms (4.6) and (4.10) is due only to the non-negativity constraint.

On the contrary, in the Poisson case, the functional  $D(x)$ , although convex, is not quadratic. Thus, the non-linearity of RLA and related algorithms is due to the particular form of  $D(x)$  as well as to the constraint used here.

(3) For all the algorithms, we set the initial estimate to a constant value

$$x_i^{(0)} = \frac{\sum_{j=1}^N y_j}{N} \forall i \quad (5.1)$$

where  $N$  is the number of pixels in the image; moreover, the integral of the image is generally normalized to 1. We must observe that in RLA we have for all  $k$ ,  $\sum_i x_i^{(k)} = \sum_i y_i$ . This is due to the fact that the functional  $D(x)$  implicitly contains the constant intensity constraint  $\sum_i [(Hx)_i - y_i] = 0$  together with  $\sum_j h_{ji} = a_i = 1$ . This later property appears naturally for the PSF in a deconvolution problem or can be imposed in the algorithm by means of the normalization factor  $a_i$ .

For ISRA, such properties do not exist, so that normalization must be performed after each iteration.

(4) In algorithms (3.4) and (4.10), the corrective term added to the current solution value at each

iteration is in the two cases of the form of a relative error, that is

For (3.4)

$$x_i^{(k)} \left\{ H^T \left[ \frac{y - Hx^{(k)}}{Hx^{(k)}} \right] \right\}_i, \quad (5.2)$$

For (4.10)

$$x_i^{(k)} \left\{ \left[ \frac{H^T R y - H^T R H x^{(k)}}{H^T R H x^{(k)}} \right] \right\}_i. \quad (5.3)$$

In the first case, the relative error vector is computed first then multiplied by  $H^T$ , while the reverse order occurs in the second case.

On the contrary, in (4.6), the corrective term is

$$x_i^{(k)} [H^T R y - H^T R H x^{(k)}]_i. \quad (5.4)$$

It has the form of an absolute difference. Comparing (4.6) and (4.10), we can see that, if the relaxation factor  $\alpha_i^{(k)}$  is not considered, the weighting factors applied to the components of the negative gradient have very different numerical values. We will show in the following section, that the speed-up effect due to the relaxation factor is much more important in (4.6) than in (4.10), for all the examples we have considered.

(5) In the general form (2.4), the global direction of displacement  $d^{(k)}$  (2.8) is a descent direction since all the terms multiplying the negative gradient are positive.

The term  $\text{diag}[x_i^{(k)}]$  is due to the non-negativity constraint; this remark can be extended to other hard constraints (paper in preparation). We want to emphasize the particular role of the function  $f_i(x)$  (cf. Section 2). In the Gaussian case, the particular expression of  $f_i(x)$  (4.9) allows proposing the algorithm in the form (4.12). This form is very close to that obtained directly from the Poisson process (3.5) with a constant value for  $f_i(x)$ . With such a choice, pure multiplicative forms are easily obtained making  $\alpha_i^{(k)} = 1 \forall i, k$ . In this case, the non-negativity of the solution is ensured, and the convergence was demonstrated [13,34].

The exact expression of  $f_i(x)$  clearly depends on the function  $g(x)$  expressing the constraints and on the objective function  $J(x)$ . This is not obvious when specific expressions of the negative gradient are used. For a particular choice of  $f_i(x^{(k)})$ , the

descent direction at the current iteration step is completely fixed, and the only possible acceleration method is to optimize the descent stepsize in the limits imposed by the constraints.

However, several algorithms suggested in the literature use an implicit modification of the function  $f_i(x^{(k)})$  to increase the speed of the algorithm. Among such attempts, we will mention that of Zaccheo and Gonzalvès [44] in which the non-negativity of the solution is imposed by the variable change  $x = u^2$ . We will show in the next section, how these algorithms can be obtained by changing in our method the function  $g(x)$  expressing the constraints and, as a consequence, a change of the function  $f(x)$ .

(6) These algorithms are not regularized. The non-negativity constraint, although necessary, does not produce smooth solutions. When the iteration number increases too much, the classical phenomenon of noise amplification appears in the solution. From a practical point of view, the iteration number must be limited to obtain an acceptable compromise between resolution and stability of the solutions; this allows some regularization.

In the present paper, we use simulated data to test the algorithms, then a direct comparison of the restored image with the true image allows us to determine the optimal iteration number.

Recently [24–26], we proposed to use a Wiener filter as a reference to determine the optimal iteration number in the restoration process or to determine the optimal regularization factor if explicit smoothness regularization is introduced. In [33], the Wiener filter is used in a pre-filtering procedure before the regularized restoration. However, the regularization is out of the scope of the present paper, this point has been in part presented in a recent communication [24] and will be analyzed more deeply in a future paper.

## 6. Influence of the function $g(x)$ . Accelerated algorithms

In the previous sections, the non-negativity constraint is expressed using the function  $g(x) = x$ , we show now that other functions can be used to this end and that in so doing, we can obtain other

multiplicative algorithms with higher convergence rates.

Let us consider, for example, the case where the non-negativity constraint is expressed using the general function  $g(x) = x^{1/n}$ . In the Poisson as well as in the Gaussian case, if we denote  $U(x^{(k)})$  and  $V(x^{(k)})$ , two positive functions  $\forall x^{(k)}$ , we can write

$$-\nabla J(x^{(k)}) = U(x^{(k)}) - V(x^{(k)}). \tag{6.1}$$

The KT condition (2.2e) writes

$$x_i^{1/n}[U(x) - V(x)]_i = 0. \tag{6.2}$$

After some simple algebra detailed in Appendix A, we obtain the general form of the accelerated algorithm. The application to the Poisson and to the Gaussian noise cases are summarized as follows.

### 6.1. Poisson case

Denoting

$$U(x) = H^T \frac{y}{Hx} \quad \text{and} \quad V(x) = a_i, \tag{6.3}$$

we have from (A.4):

$$f_i(x^{(k)}) = \frac{1}{a_i^n} \left[ \sum_{m=0}^{n-1} a_i^m \left( H^T \frac{y}{Hx^{(k)}} \right)_i^{n-1-m} \right], \tag{6.4}$$

instead of  $f_i(x^{(k)}) = 1/a_i$  as in Section 3.

The relaxed algorithm deduced from relation (2.4) or (A.3), will be

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} \frac{x_i^{(k)}}{a_i^n} \left[ \sum_{m=0}^{n-1} a_i^m \left( H^T \frac{y}{Hx^{(k)}} \right)_i^{n-1-m} \right] \times \left[ \left( H^T \frac{y}{Hx^{(k)}} \right)_i - a_i \right]. \tag{6.5}$$

From (A.7), the maximum step size for non-negativity is

$$\alpha_P^{(k)} = \text{Min}/i \frac{a_i^n}{a_i^n - (H^T(y/Hx^{(k)}))_i^n} \quad \forall i, \tag{6.6}$$

such that  $[\nabla D(x)]_i > 0$  and  $x_i > 0$ .

The value of the optimal stepsize  $\alpha_C^{(k)}$  must be computed by a line search procedure in the range  $]0, \alpha_P^{(k)}]$  in the direction deduced from (A.8):

$$d^{(k)} = \text{diag} \left\{ \frac{x_i^{(k)}}{a_i^n} \left[ \sum_{m=0}^{n-1} a_i^m \left( H^T \frac{y}{Hx^{(k)}} \right)_i^{n-1-m} \right] \right\} \times \left[ \left( H^T \frac{y}{Hx^{(k)}} \right)_i - a_i \right]. \tag{6.7}$$

The descent direction as well as the magnitude of the descent vector are modified in comparison with the basic algorithm (3.5). Close to the convergence, the magnitude is multiplied by “ $n$ ”.

In the unrelaxed case, that is  $\alpha_C^{(k)} = 1 \forall k$ , we obtain  $\forall i$ , a multiplicative algorithm:

$$x_i^{(k+1)} = \frac{x_i^{(k)}}{a_i^n} \left( H^T \frac{y}{Hx^{(k)}} \right)_i^n. \tag{6.8}$$

In the particular case of the deconvolution problem we have also  $a_i = 1 \forall i$ .

The origin of the algorithms proposed by Zaccaro and Gonsalves [44] and by Llacer and Nuñez [27] is then fully evidenced by our method as well as the basis of the technique they use to speed-up the algorithms. The convergence of algorithm (6.8) was not demonstrated, but in all examples shown in the following section, we do not observe any divergence in the case  $n = 2$ . For  $n > 2$ , to avoid divergence problems, the general relaxed form (6.5) can be used.

### 6.2. Gaussian case

If the data “ $y$ ” are such that  $H^T R y \geq 0$  (if such is not the case, the data must be shifted toward positive values), we have

$$U(x) = H^T R y, \quad V(x) = H^T R H x \tag{6.9}$$

and

$$f_i(x^{(k)}) = \frac{1}{(H^T R H x^{(k)})_i^n} \times \left[ \sum_{m=0}^{n-1} (H^T R H x^{(k)})_i^m (H^T y)_i^{n-1-m} \right]. \tag{6.10}$$

The relaxed algorithm deduced from (2.4) or (A.3) writes

$$\begin{aligned}
 x_i^{(k+1)} &= x_i^{(k)} + \alpha_i^{(k)} \frac{x_i^{(k)}}{(H^T R H x^{(k)})_i^n} \\
 &\times \left[ \sum_{m=0}^{n-1} (H^T R H x^{(k)})_i^m (H^T y)_i^{n-1-m} \right] \\
 &\times [H^T R y - H^T R H x^{(k)}]_i. \tag{6.11}
 \end{aligned}$$

The maximum stepsize for non-negativity is then

$$\alpha_p^{(k)} = \text{Min}_i \frac{1}{1 - (H^T R y / H^T R H x^{(k)})_i^n} \quad \forall i, \tag{6.12}$$

such that  $[\nabla G(x)]_i > 0$  and  $x_i > 0$ .

The value of  $\alpha_C^{(k)}$  must be computed by a line search procedure in the range  $]0, \alpha_p^{(k)}]$  in the direction

$$\begin{aligned}
 d^{(k)} &= \text{diag} \left\{ \frac{x_i^{(k)}}{(H^T R H x^{(k)})_i^n} \right. \\
 &\times \left. \left[ \sum_{m=0}^{n-1} (H^T R H x^{(k)})_i^m (H^T y)_i^{n-1-m} \right] \right\} \\
 &\times \left[ \frac{H^T R y}{H^T R H x^{(k)}} - 1 \right]. \tag{6.13}
 \end{aligned}$$

The descent direction used in ISRA (4.14) has been changed as well as the magnitude of the descent vector. The use of function (6.10) allows obtaining in the unrelaxed case the multiplicative algorithm, indeed if  $\alpha_C^{(k)} = 1 \forall k$ , we have  $\forall i$

$$x_i^{(k+1)} = x_i^{(k)} \left( \frac{H^T R y}{H^T R H x^{(k)}} \right)_i^n. \tag{6.14}$$

As in the Poisson case, the convergence of this algorithm was not demonstrated, but with  $n = 2$ , no divergence was observed in our examples. To avoid divergence problems when  $n > 2$ , the general relaxed form (6.11) can be used.

In any case, the proposed method shows that using different forms of the function  $g(x)$  to express the same constraints, we can reach a large class of algorithms. Writing the algorithms in a modified gradient form, we fully justify the method suggested in the literature to accelerate the basic algorithms [27,44]. Indeed when we are close to the convergence, the modulus of the descent stepsize is multi-

plied by “ $n$ ” when compared to the basic algorithms, independently of the descent direction. For all the images considered here, the convergence occurs when  $n = 2$ , with an acceleration factor of 2. Unfortunately, the multiplicative algorithms (6.8) and (6.14) diverges when  $n \geq 3$ ; however, in any case, the convergence can be ensured using the relaxed forms, with an appropriate computation of the stepsize by line search. In the following section we give only results in the unrelaxed cases for  $n = 2$  and we left for a future work a complete analysis of the relaxation of these algorithms.

### 7. Numerical illustrations

We show in this section the results given by the proposed algorithms in the case of an image deconvolution problem. Our objective is to analyze the behavior of the relaxed and “accelerated” algorithms and to show in which situations an important speed increase can be observed.

Two test images were used to check the algorithms in two opposite situations; one is an astronomical simulated image of  $64 \times 64$  pixels, chosen to represent an ensemble of three stars, with different diameters and centers to limbs variations. A bright and dark sunspot-like feature is added to the larger star’s atmosphere; the features of this image are the high range of intensity values and the large number of zeros. The second image is the image of LENA sampled on  $128 \times 128$  pixels apodized near the edges by a Gaussian function. It is a smooth image with no zero values. Two PSFs are used to blur these images, one is narrow to simulate moderate blur, and the other is larger to simulate severe blur. They are realistic representations of the PSF of a true telescope; indeed, they are computed as the squared modulus of the Fourier transform of a function representing the telescope aperture with a small phase aberration. They may correspond to observing conditions with the Hubble Space Telescope operating in the far ultraviolet. The resulting optical transfer functions are then low-pass filters, limited in spatial frequencies to the extent of the aperture auto-correlation function. The observed filtered images are then strictly band limited. While these PSFs are evidently

adapted to the astronomical simulated image, they are also used to blur the image of Lena.

The corresponding data are represented in Figs. 1 and 2, for the two images.

The blurred signals are corrupted either by a Poisson or by a Gaussian noise. For each blurred image, three levels of noise are considered. From a rigorous point of view, given a specific noise (Gaussian or Poisson), a different algorithm should be used; however, we will show that within certain limits, the result of the restoration is only weakly dependent on the algorithm used.

### 7.1. Images corrupted by a Poisson noise

The blurred images corrupted by Poisson noise, are represented in Figs. 3 and 4 and constitute the data to be deconvolved by the different algorithms. The noise NS1 corresponds to  $10^8$  photons in the images, that is to a very low noise level, the noise NS2 corresponds to  $10^6$  photons in the images, while noise NS3 corresponds to only  $10^4$  photons in the images, that is, to very noisy data.

The results obtained with ISRA and RLA algorithms in their different versions (basic, relaxed or accelerated, see Appendix C) are summarized

in Table 1. The iteration numbers given in this table corresponds to the “best” image, that is to the image for which the Euclidean distance  $\|x^{(k)} - x^*\|^2$  is minimum (where  $x^*$  stands for images 1a or 1b). For a given set of data, corresponding to a row of Table 1, we use the same noise realization, and then we have the same noisy blurred image.

#### 7.1.1. General remarks

The analysis of Table 1 leads us to different conclusions:

*Influence of the strength of the blur:* For a given image, the iteration number is always greater in the case of a severe blur, which is not surprising.

*Influence of the algorithm (ISRA or RLA) used:* The number of iterations for ISRA is always greater than or equal to that for RLA. The gain on the iteration number is generally higher (or approximately the same) for RLA than for ISRA.

*Influence of the noise level:* When the noise level increases, the iteration number corresponding to the best image, decreases. The effectiveness of the relaxation procedure is small in the case of very noisy data.

*Influence of the type of the image:* For identical blur and noise level, the astronomical simulated image, that is a strongly contrasted image, requires

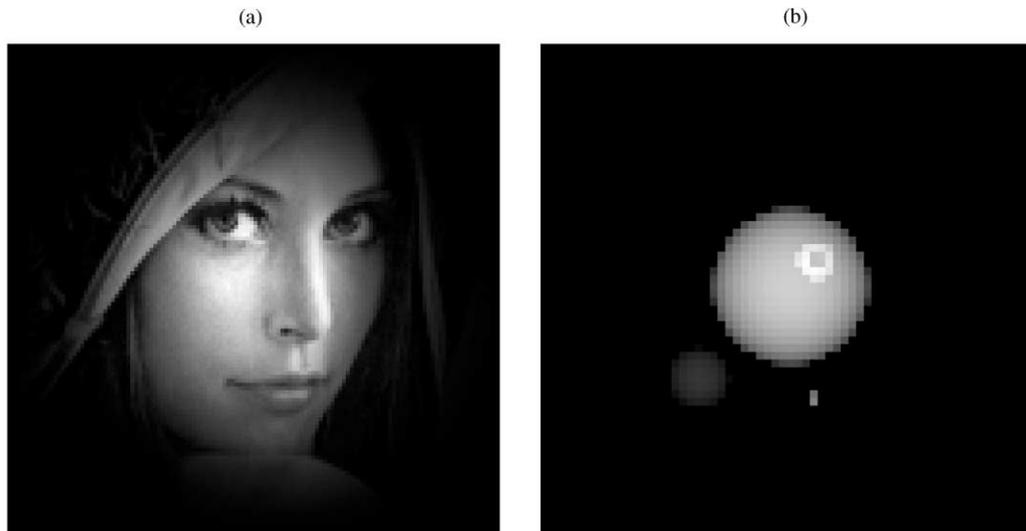


Fig. 1. Original images used in the numerical simulation (denoted  $x^*$  in the paper). Left:  $128 \times 128$  pixels image of Lena, apodized with a Gaussian function. Right:  $64 \times 64$  pixels image of a simulated astronomical object representing an ensemble of three stars, with various center to limb variations and a bright structure on the larger star.

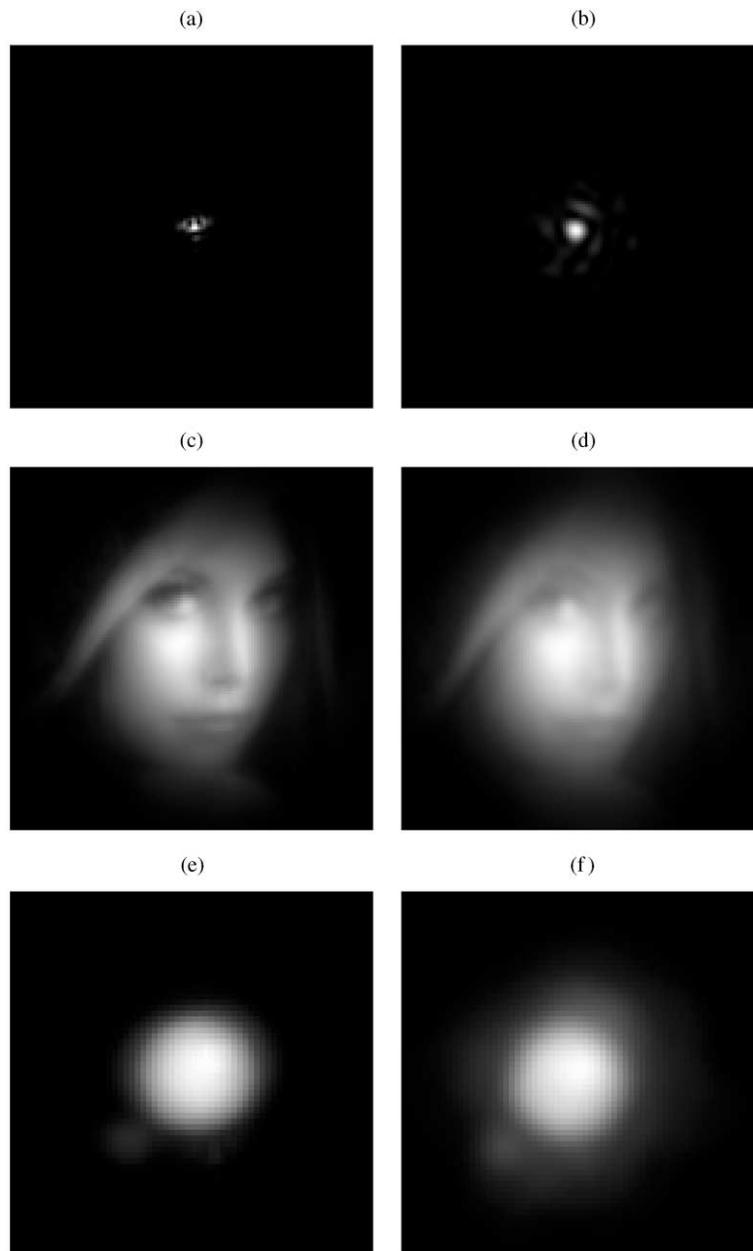


Fig. 2. Images of Fig. 1 degraded by a moderate (left) and a severe (right) blur. From left to right and top to bottom: PSFs used for blurring the images, noiseless blurred images of Lena and noiseless blurred images of the astronomical object.

in almost all case, more iterations than the image of Lena. For RLA, the gain is generally more important for the simulated image than for the Lena image; the opposite effect occurs for ISRA, indeed,

for the simulated image, the relaxation of ISRA is inefficient (gain about 1).

*Other remarks:* The highest gains appear for RLA in the case of the simulated image, for a severe blur,

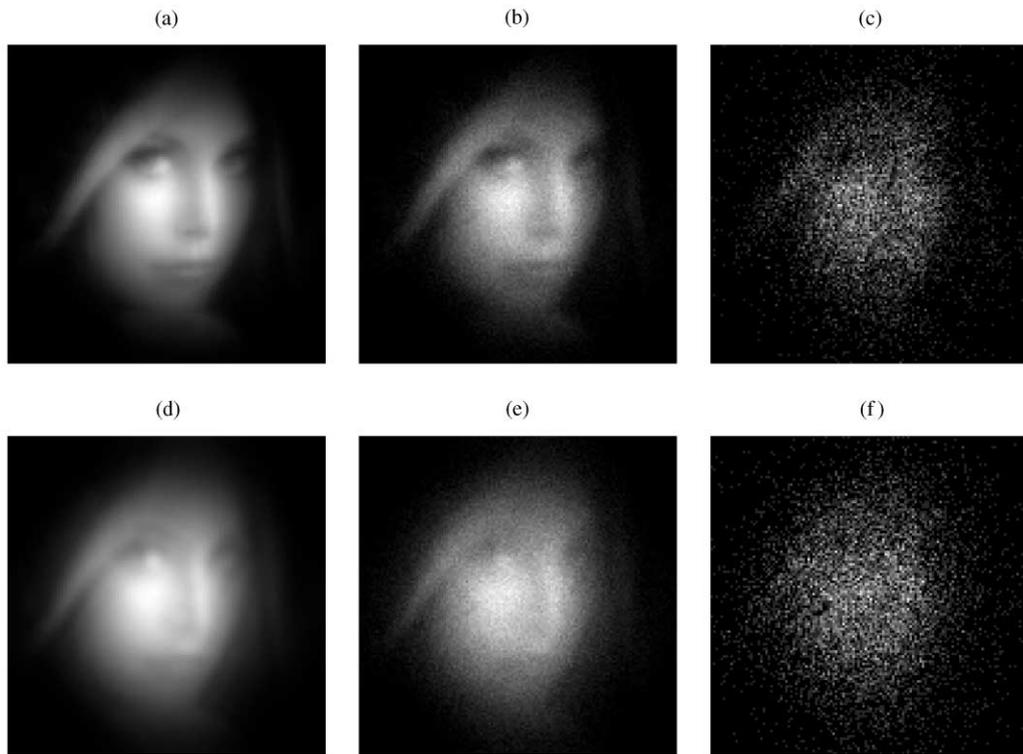


Fig. 3. Poisson statistics. Simulated noisy images of Lena. Top: moderately blurred image (Fig. 2c) observed with  $10^8$  (a),  $10^6$  (b) and  $10^4$  (c) photoelectrons per image. Bottom: severely blurred image (Fig. 2d) observed with  $10^8$  (d),  $10^6$  (e) and  $10^4$  (f) photoelectrons per image.

for low or intermediate noise levels (rows 13 and 14 in Table 1).

In the case of accelerated algorithms with  $n = 2$ , the behavior is similar, whatever the origin of the algorithm (Gauss or Poisson). The gain factor is 2 in almost all cases. The only exception is for the noisy Lena image moderately blurred. In the noisier case (NS3), these algorithms are slower than the basic versions; this is probably because in these algorithms, the term  $H^T y$  appears in the descent direction. This shows that such algorithms do not permit in all cases a speed increase.

#### 7.1.2. Effects of various realizations of the noise process

To avoid erroneous conclusions due to a unique realization of the noise process, we have considered in some typical cases of Table 1, an ensemble of realizations for the same amplitude of the noise.

(a) *Lena image*: In the case of the Lena image severely blurred, with an intermediate noise level NS2 (row 5 of Table 1 and Fig. 3e), the conclusions are the following:

For ISRA, the iteration numbers for the best images are in the range of  $\pm 10\%$  of the values indicated in Table 1, while the dispersion of the gain factor is  $\pm 3\%$  for the relaxed algorithm. There is practically no spread of the gain factor for the accelerated algorithm.

For RLA, the results are similar in which concerns the dispersion of the iteration number, while the spread of the gain value is about  $\pm 10\%$  for the relaxed algorithm. The observations concerning the accelerated algorithm are the same as that for ISRA.

(b) *Simulated data*: We proceed in the same way for the astronomical-type simulated image severely blurred with the noise level NS2 (row 14 of

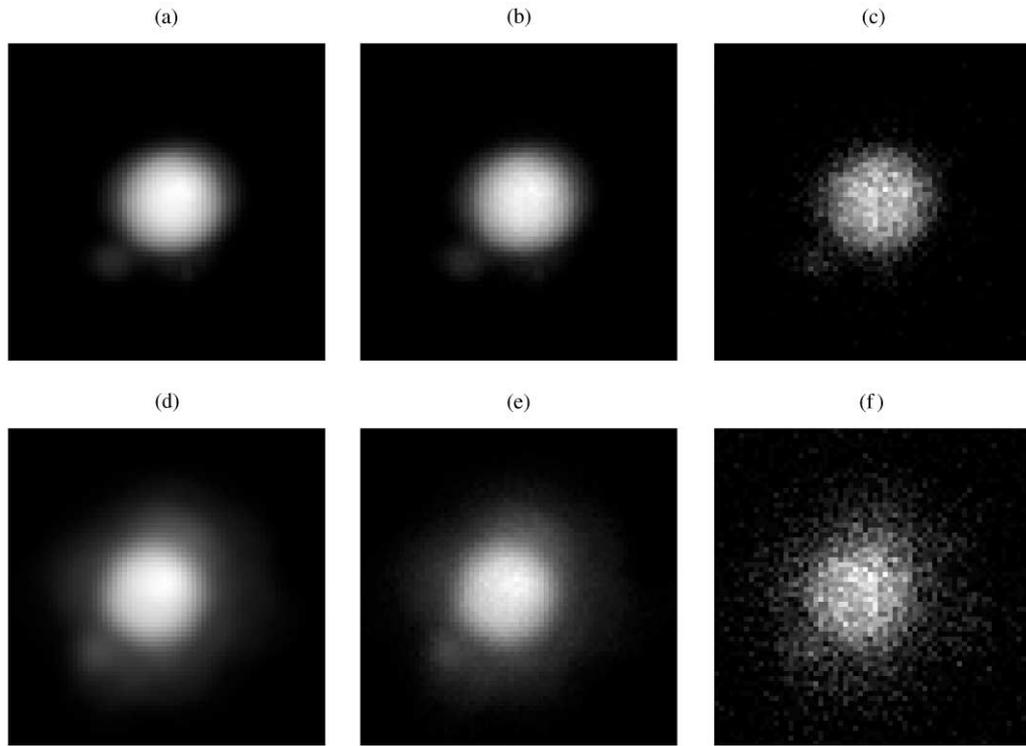


Fig. 4. Poisson statistics. Simulated noisy images of the astronomical object. Top: moderately blurred image (Fig. 2e) observed with  $10^8$  (a),  $10^6$  (b) and  $10^4$  (c) photoelectrons per image. Bottom: severely blurred image (Fig. 2f) observed with  $10^8$  (d),  $10^6$  (e) and  $10^4$  (f) photoelectrons per image.

Table 1

Row number	Image	PSF	Noise (NS)	ISRA						RLA					
				Basic		Relaxed		Accelerated		Basic		Relaxed		Accelerated	
				Ite. nb.	Gain	Ite. nb.	Gain	Ite. nb.	Gain	Ite. nb.	Gain	Ite. nb.	Gain	Ite. nb.	Gain
1	Lena image	Narrow	1	189	65	2.9	94	2	188	89	2.1	94	2		
2			2	13	6	1.6	11	1.2	13	7	1.9	11	1.2		
3			3	2	2	1	3	0.7	2	2	1	3	0.7		
4		Large	1	1507	467	3.2	753	2	1137	456	2.5	568	2		
5			2	38	11	3.4	19	2	39	16	2.4	19	2		
6			3	4	3	1.3	2	2	3	3	1	3	1		
7		Very large	1	9842	3397	2.9	4920	2	5746	1792	3.2	2873	2		
8	2		418	138	3	209	2	280	89	3.2	140	2			
9	3		9	5	1.8	4	2.2	10	3	3.3	5	2			
10	Simulated data	Narrow	1	2743	2725	1	1371	2	1570	558	2.8	785	2		
11			2	81	76	1.1	40	2	82	27	3	40	2.1		
12			3	6	5	1.2	3	2	6	5	1.2	3	2		
13		Large	1	54636	54597	1	27319	2	14420	2334	6.2	7210	2		
14			2	634	625	1	317	2	295	42	7	147	2		
15			3	17	15	1.1	8	2	16	6	2.6	8	2		

Table 1 and Fig. 4e), and the results are rather different:

For RLA, we show in Figs. 5a and b, respectively, the histograms of the optimal iteration number and of the acceleration factor, for various realization of the noise at the same level. The relaxation effect is particularly important in the case of the simulated astronomical image severely blurred with a very low noise, but we have not obtained the high-gain values mentioned by White [42].

For ISRA, the dispersion of the iteration number for the best image is  $\pm 18\%$ , for the basic as well as for the relaxed algorithm. The gain factor is sharply peaked to 1 for the relaxed algorithm (no relaxation effect) and is always 2 for the accelerated.

7.1.3. Analysis of the reconstruction error and of restored images

(a) *Lena image*: In Figs. 6a and b, we show typical reconstruction error curves, respectively, for ISRA

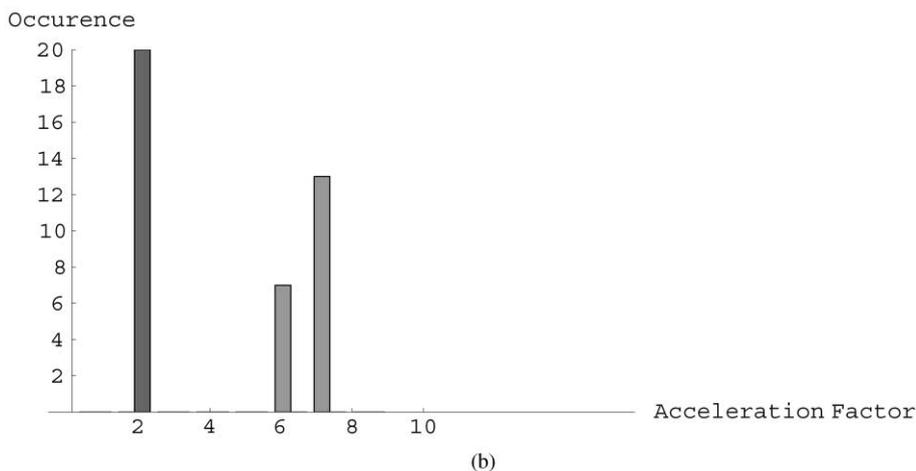
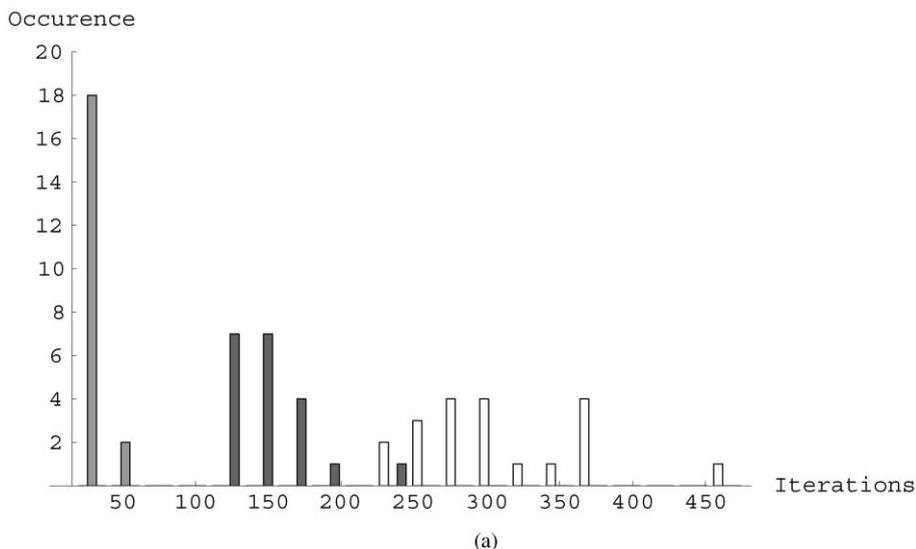


Fig. 5. Histograms of the various optimal iteration numbers (a) and acceleration factors (b) obtained with the RLA algorithm, for 20 different noise realizations of the severely blurred astronomical object (see Table 1, row 14). Bar gray levels correspond to different versions of the algorithm: accelerated (black), relaxed (gray) or raw (white). The iteration number is  $284 \pm 25$  for the basic algorithm, while it is  $40 \pm 5$  in the relaxed case (mean gain about 7). For the accelerated algorithm, the gain is 2, with a very low spread.

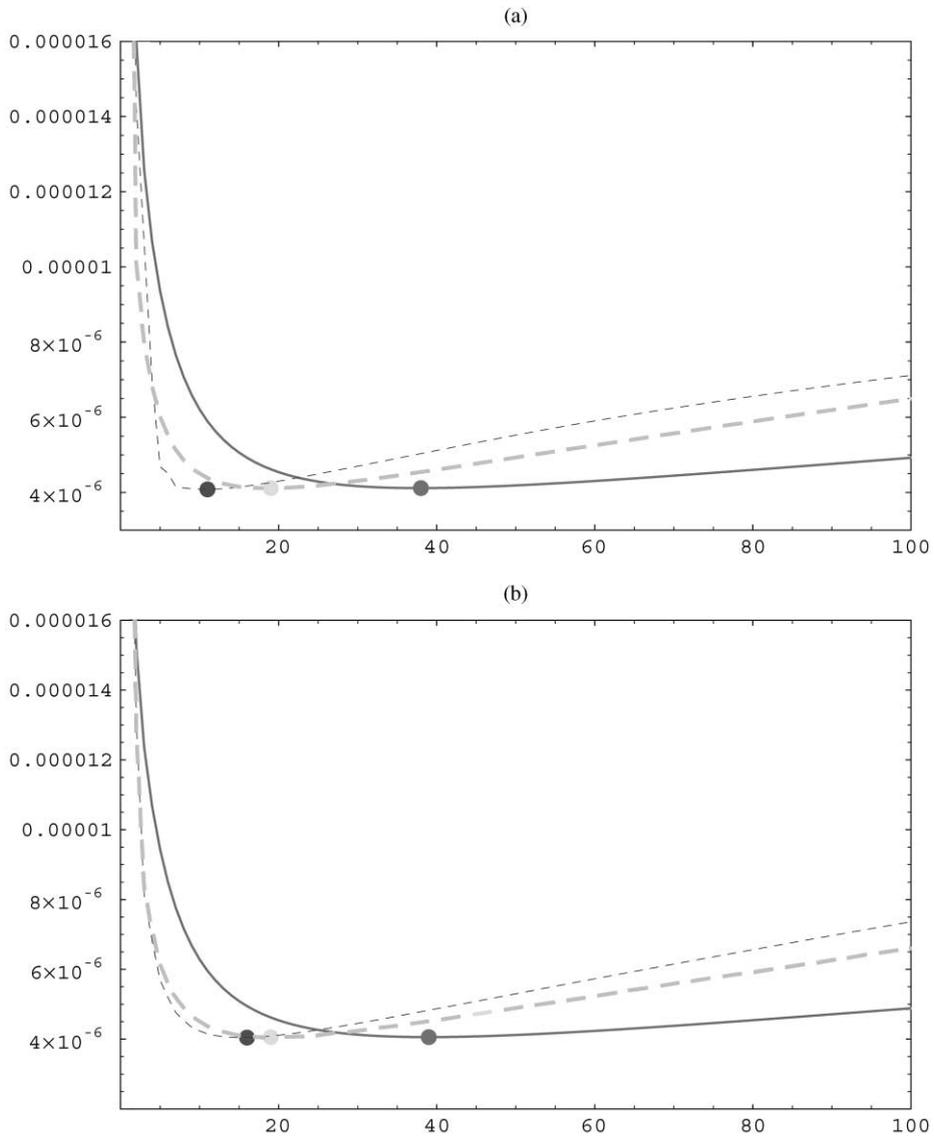


Fig. 6. Reconstruction error  $\|x^* - x^{(k)}\|^2$  as a function of the iteration number for ISRA (top) and RLA (bottom) in the basic (continuous line), relaxed (thin dashed line) and accelerated (thick dashed line) versions of the algorithms. Data are those of Fig. 3e. Dots indicate the positions of minima (see Table 1, row 5 for numerical values).

and RLA. They correspond to the Lena image severely blurred, corrupted by a noise with an intermediate level NS2 (row 5 of Table 1 and Fig. 3e). Analogous curves are always observed, with relative positions deduced from Table 1. In any case, after the initial decrease, the minimum is reached, and when the iteration number increases, the re-

construction error grows, due to the noise amplification in the non-regularized reconstruction process. This effect is clearly more pronounced when the noise level increases. The minimum values of the reconstruction error are very close as well as the best reconstructed images, whatever the algorithm used.

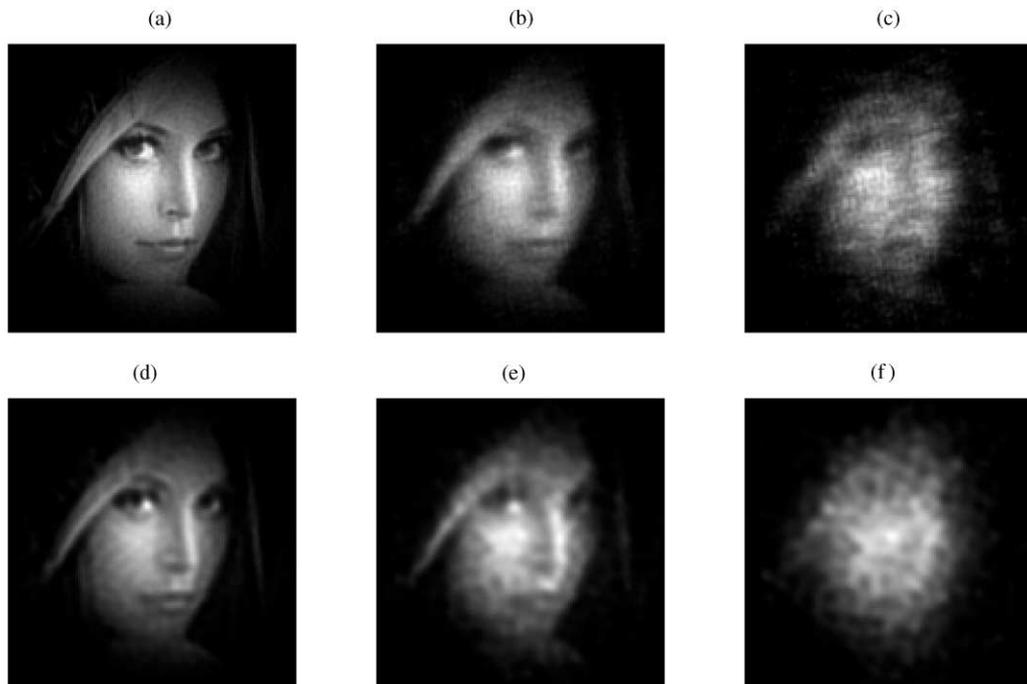


Fig. 7. Images a–f are the results of the deconvolution of corresponding images (a–f) of Fig. 3. From a given starting image, the results are almost the same whatever the version of the algorithm, and even whatever the algorithm used. The optimal iteration numbers obviously differ from a computation to another (see Table 1, rows 1–6 for numerical values).

We show in Fig. 7, the best images obtained for different noise levels (Table 1, rows 1–6, and Fig. 3). Evidently, the reconstruction is better for low noise levels, and the result is not convincing for the highest noise, but it is not surprising if we consider the aspect of the corresponding data in Figs. 3c and f.

(b) *Simulated data*: We consider in Figs. 8a and b the curves of the reconstruction error for the astronomic-type simulated images severely blurred, corrupted by an intermediate noise NS2 (row 14 in Table 1 and Fig. 4e). For a comparison with Figs. 6a and b, only the basic image is changed. For such image with many zeros, the relaxation of ISRA is ineffective, while this effect is important for RLA.

We show in Fig. 9 the best reconstructed images for the data shown in Fig. 4 and Table 1, rows 10–15. For a given set of data, the restored images are very close for ISRA and RLA. The minimum values of the reconstruction error are also very close for both the algorithms.

#### 7.1.4. Influence of the noise level and the blur extent on the reconstruction

*Noise level*: The influence of the noise level on the reconstruction error is shown in Fig. 10 for ISRA. The corresponding cases are those of rows 4–6 in Table 1; clearly, when the noise level increases, the minimum reconstruction error increases and the optimal iteration number decreases. These curves are typical examples, and an analogous behavior is observed with a different PSF and other images. Similar results are observed with RLA.

*Blurring*: For a better understanding of the effect of the PSF, we use the PSF represented in Fig. 11a, to blur the image of Lena. This PSF is different and larger than the previous ones. The blurred image is given in Fig. 11b, and the data are shown in Figs. 12a–c for the three noise levels previously used.

The results for the optimal iteration numbers and the acceleration factors are given in Table 1, rows 7, 8, 9 and can be compared with rows 4, 5, and 6.

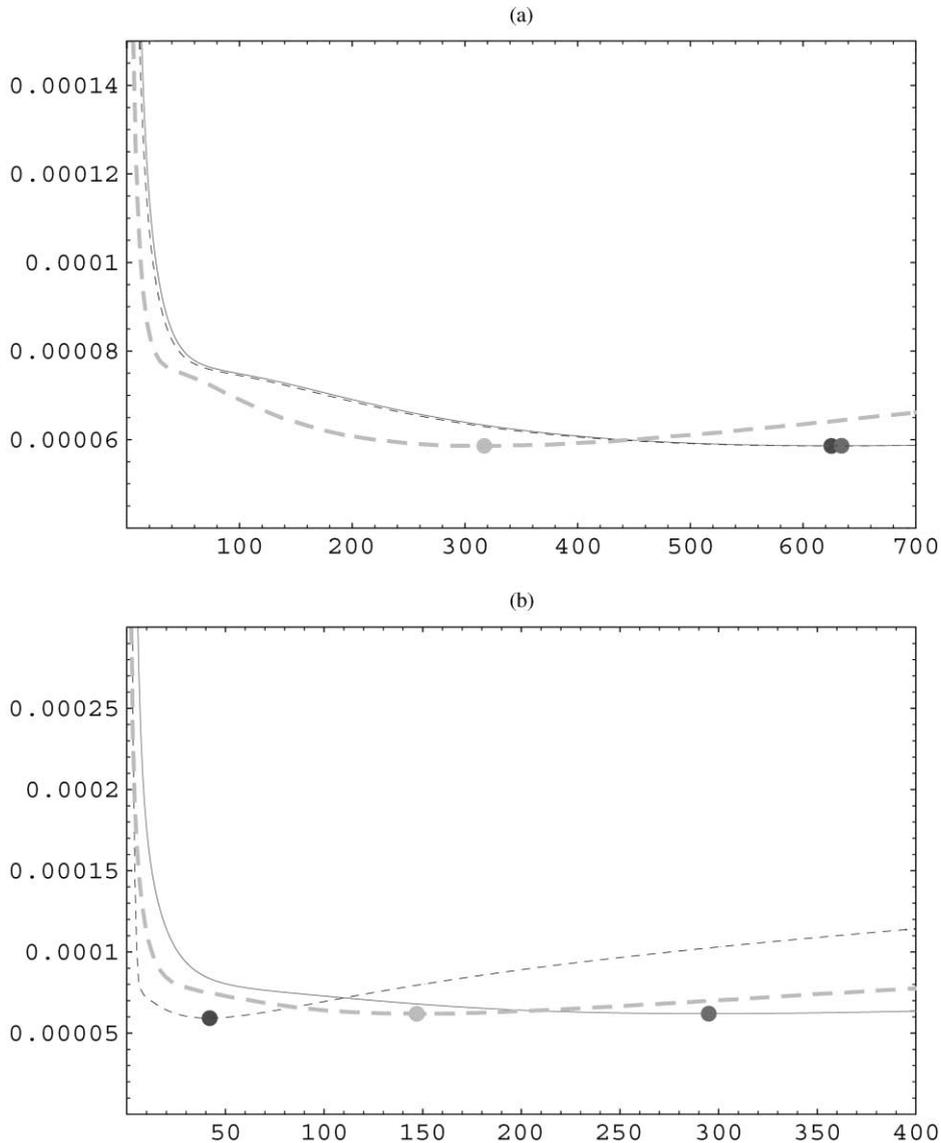


Fig. 8. Reconstruction error  $\|x^* - x^{(k)}\|^2$  as a function of the iteration number for ISRA (top) and RLA (bottom) in the basic (continuous line), relaxed (thin dashed line) and accelerated (thick dashed line) versions of the algorithms. Data are those of Fig. 4e. Dots indicate the positions of minima (see Table 1, row 14 for numerical values).

In the case of an intermediate noise level NS2 (row 8 and Fig. 12b), for an ensemble of realizations of the noise, the dispersion of the results (not shown in Table 1) leads to the following comments:

For ISRA, the dispersion of the iteration number for the best image is now  $\pm 50\%$  for all the versions of the algorithm. This effect is also observed for gain factors whose dispersion is about  $\pm 16\%$  for

the relaxed algorithm while for the accelerated algorithm the gain is always 2 with an extremely low spread.

For RLA, the spread of the iteration number for the best image is about  $\pm 30\%$ , while the dispersion on the gain values is  $\pm 15\%$  for the relaxed algorithm. The gain is always close to 2 for the accelerated algorithm.

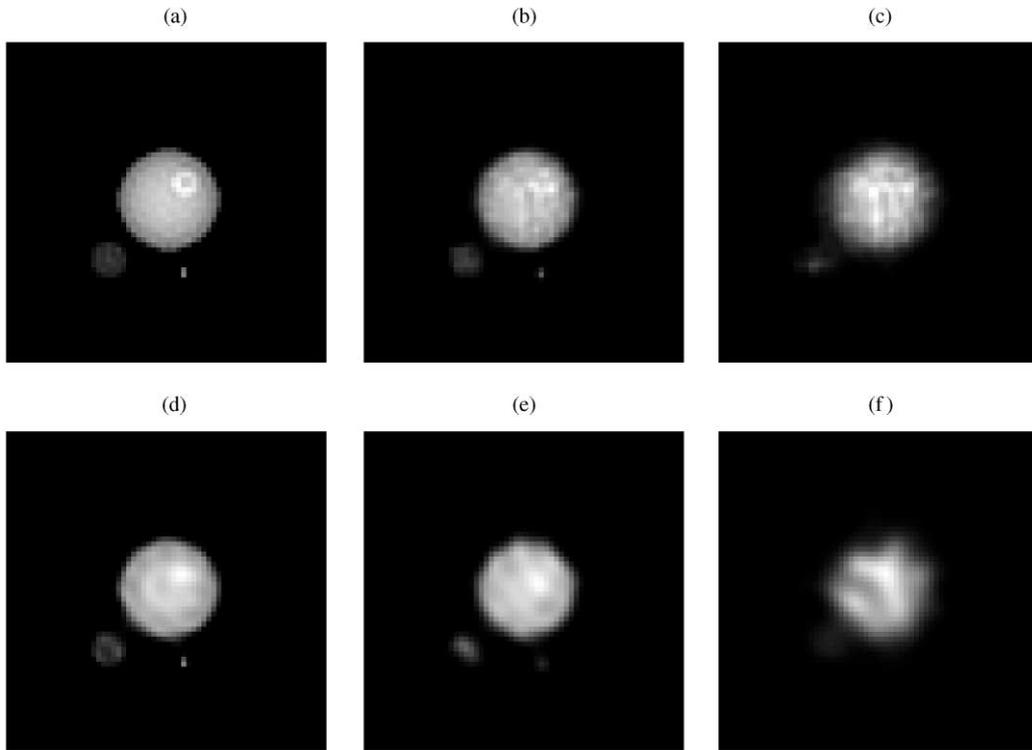


Fig. 9. Images a to f are the results of the deconvolution of corresponding images (a–f) of Fig. 4. From a given starting image, the results are almost the same whatever the version of the algorithm, and even whatever the algorithm used. The optimal iteration numbers obviously differ from a computation to another (see Table 1, rows 10–15 for numerical values).

The best results are shown in Fig. 12d, e and f. They correspond, respectively, to the data of Figs. 12a, b and c. The different versions of a given algorithm, show very close results whatever the origin of the algorithm. The optimal iteration numbers are indeed different in each case, as indicated in the table.

#### 7.1.5. The particular behavior of algorithm (4.6)

The effect of the relaxation is very important on algorithm (4.6); some considerations on this algorithm are mentioned in Section 5. We show in Fig. 13a, for the data of Fig. 3e, the error curve. In the unrelaxed case, several tens of thousands of iterations are necessary, while 147 iterations only are necessary in the relaxed case. The effect of the choice of the function  $f_i(x^{(k)})$  on the speed of the algorithm is clearly seen in Fig. 13b where the reconstruction error is shown for the relaxed form

of algorithm (4.6), for the pure ISRA algorithm, and for the relaxed version of ISRA.

The simple fact to use the function  $f_i(x^{(k)})$  leading to ISRA, gives the best image at the iteration 418. The gain in the iteration number due to this function is in itself very important, and fully justifies the remarks of Section 5. In the relaxed version of ISRA, the best image is obtained at iteration 138. The optimal iteration numbers are very close for the relaxed version of (4.6), and of ISRA. The major effect of the speed increase is then due to the modification of the modulus of the descent vector, rather than to the implicit modification of the descent direction due to  $f_i(x^{(k)})$ .

#### 7.2. Images corrupted by a Gaussian additive noise

We use in this case the images blurred by a large PSF corresponding to Figs. 2d and f. They

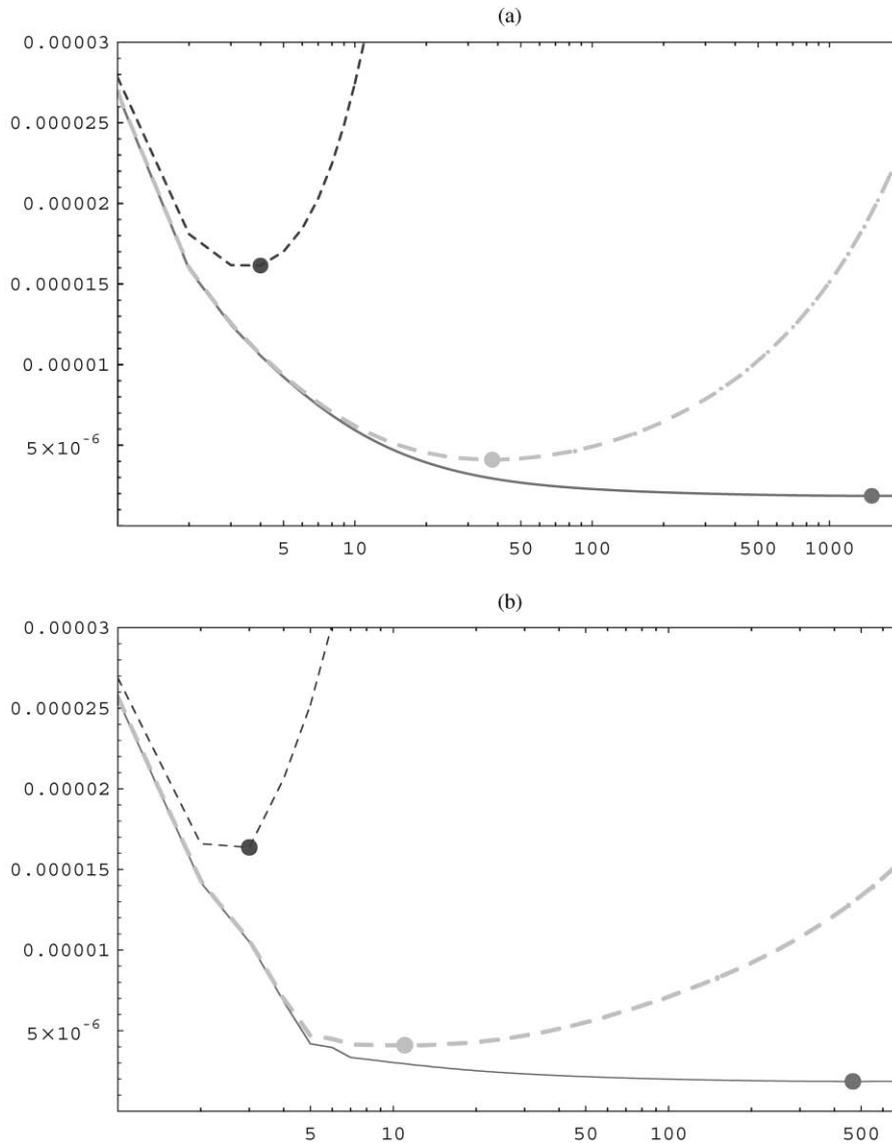


Fig. 10. Representation on a semi-logarithmic scale of the reconstruction error as a function of the iteration number for the three noises levels, for the basic (a) or relaxed (b) versions of ISRA, with noises NS1 (continuous line), NS2 (thick dashed line) and NS3 (thin dashed line). Data are those of Figs. 3d, e and f; the related parameters are found in Table 1, rows 4–6. Curves corresponding to basic, relaxed or accelerated algorithms have the same relative positions in what concern the noise. Results given by RLA are similar except for the positions of the minimum. As expected, the minimum error increases with the level of noise; images with a high SNR require a high optimal iteration number.

are corrupted by a zero mean Gaussian noise, whose standard deviation is adjusted to correspond to the intermediate noise level NS2 in the case of the Poisson noise.

The blurred noisy images are shown in Figs. 14a and b and they must be compared with Figs. 3e and 4e, respectively. When negative values appear due to the noise, the images are shifted (adding a

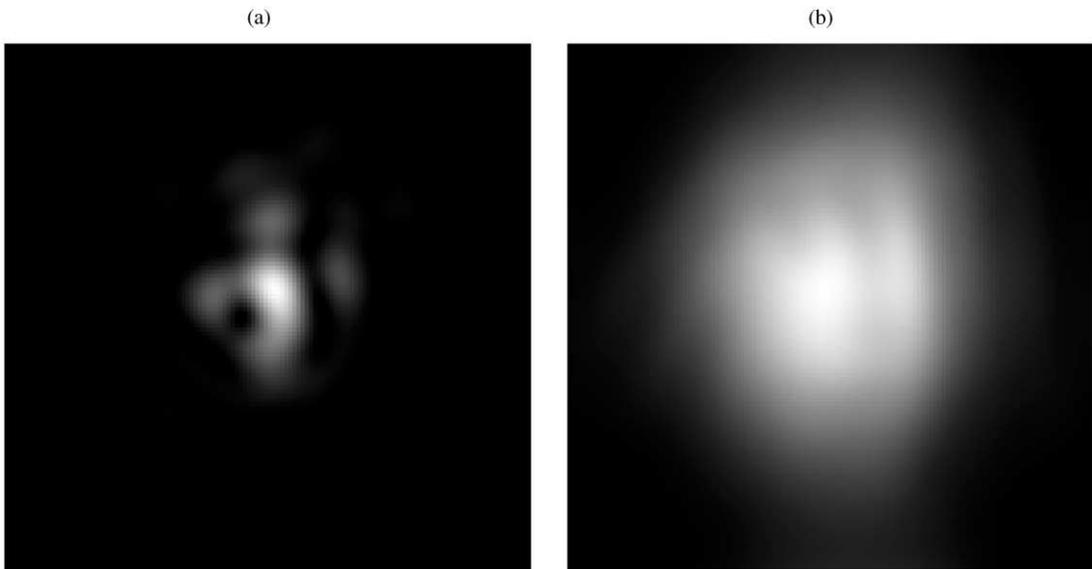


Fig. 11. Result of convolution (11b) of the image of Lena (Fig. 1) degraded by a very large PSF (11a).

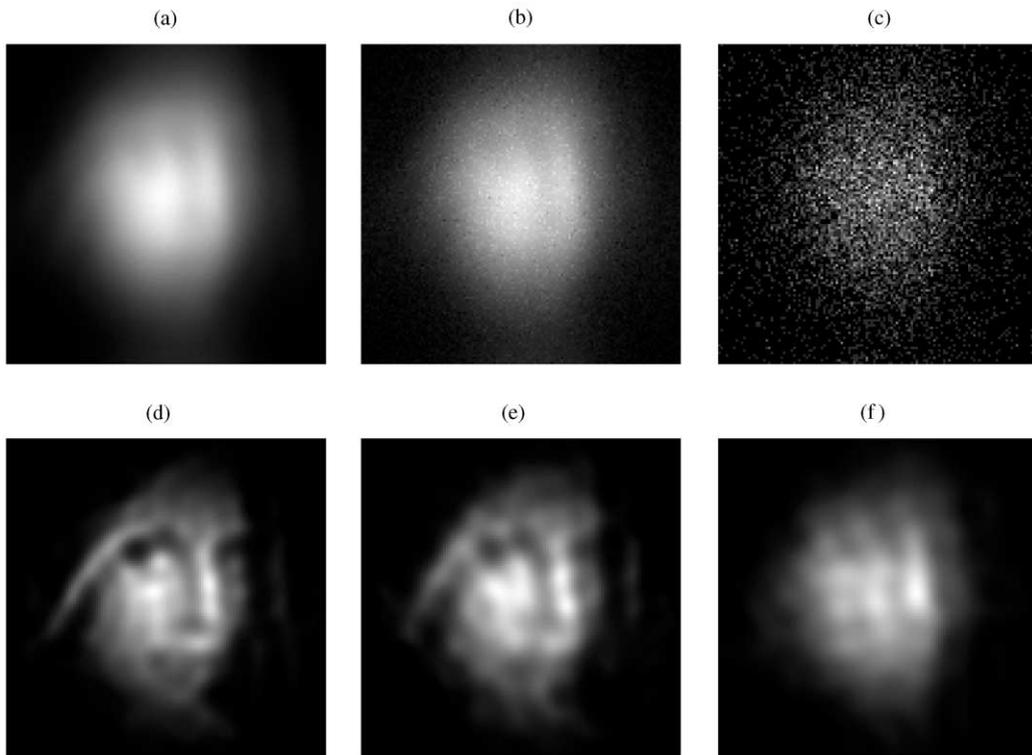


Fig. 12. Simulated noisy images of Lena (top) and best-reconstructed corresponding images (bottom). Noisy images are made of  $10^8$  (a),  $10^6$  (b) and  $10^4$  (c) photoelectrons per image. The unnoisy blurred image is that of Fig. 11b.

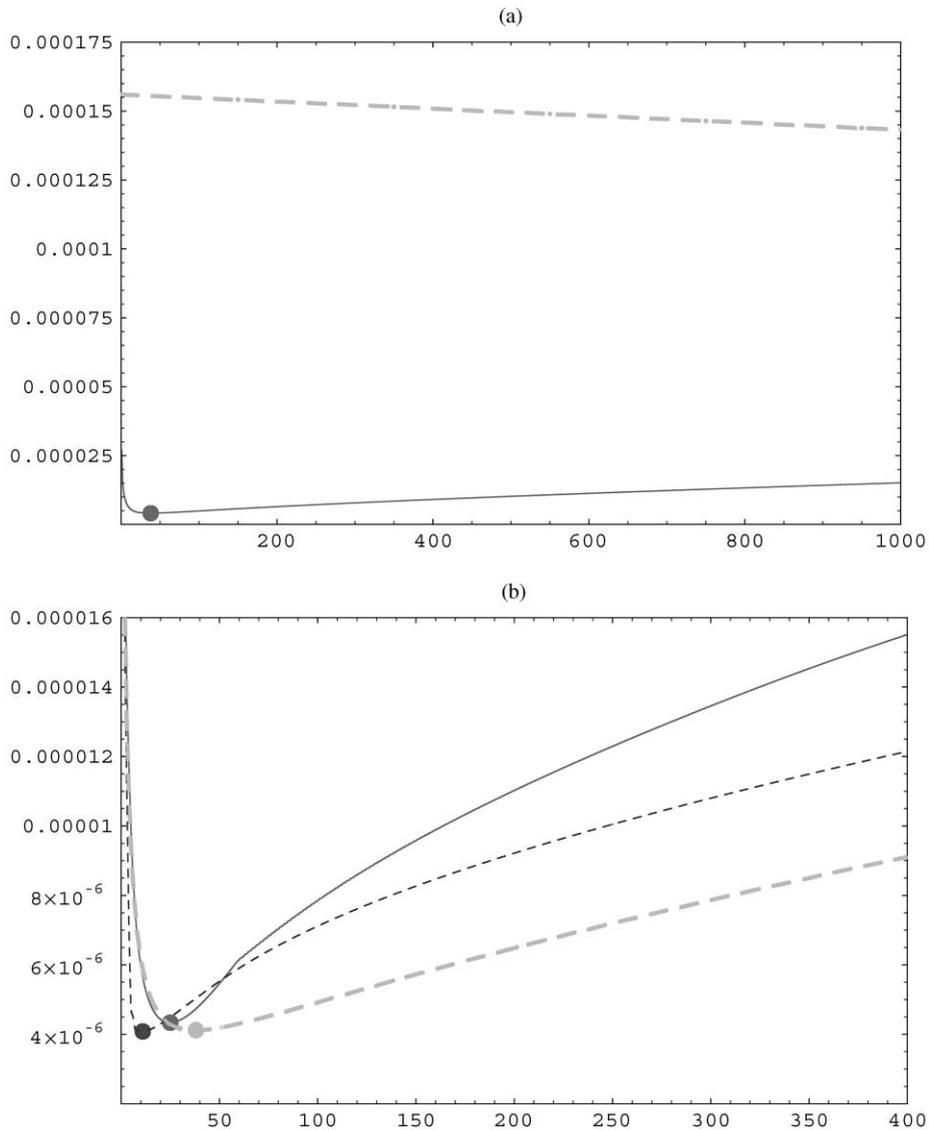


Fig. 13. (a) Reconstruction error curves obtained with the algorithm (4.6), both in its unrelaxed (very slow) version (dashed line) and in its relaxed form (continuous line). (b) the reconstruction error for the relaxed version of algorithm (4.6) is compared with that obtained for ISRA, and for the relaxed version of ISRA. The minima are obtained respectively at iterations 25, 38 and 11. Original data: Fig. 3e (row 5 of Table 1).

constant to the data), so that the data used in the algorithms are always positive or zero.

We show in Figs. 15a and b the reconstruction error curves for the data of Fig. 14a (Lena) processed, respectively, by the ISRA and RLA algorithms in their different versions. In all cases the accelerated algorithms (with  $n = 2$ ) are

faster than the basic version by a factor 2; in these two versions, RLA seems to be faster than ISRA. For the relaxed algorithms, the gain on the iteration number is 3.2 for RLA, very close to the Poisson noise case, while it is of 6.3 for ISRA (for the Poisson noise, there was no gain in such case).

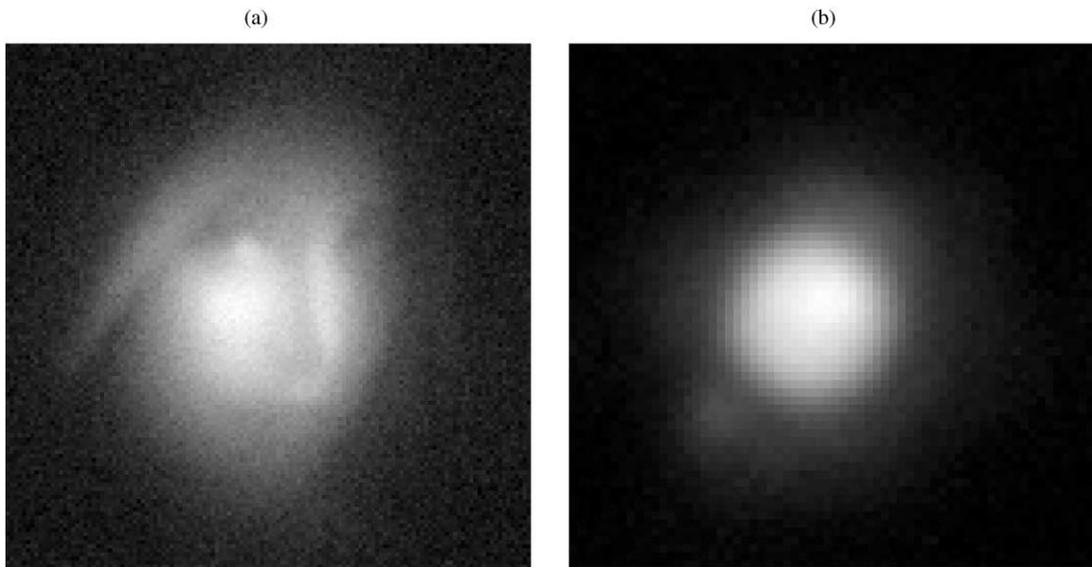


Fig. 14. Representations of severely blurred images corrupted by a Gaussian noise of variance similar to that of the NS2 Poisson noise: (a) Lena, (b) simulated astronomical object.

The analogous results for the simulated image are shown in Fig. 16a for ISRA and 16b for RLA. The accelerated algorithms exhibit as in the previous cases a gain 2 on the speed increase. Concerning the speed increase due to the relaxation, the effects are very close to those observed in the Poisson noise case, for ISRA, the gain is only 1.2, while it is about 6.1 for RLA. For a given data set, the reconstructed images are very close whatever the algorithm is used; the results are shown in Figs. 17a and b corresponding, respectively, to the data 14a and 14b.

We can then conclude that there is no fundamental difference when the two types of algorithms are used for data corrupted by the Poisson and Gaussian noise. The origin of the algorithm is not a strong constraint related to the real nature of the noise in the data.

## 8. Conclusion

We proposed in this paper a general method to devise effective multiplicative algorithms for likelihood maximization under non-negativity con-

straint; we analyze the two classical noise processes considered usually.

The originality of the proposed approach is in the fact that it can be applied to any convex function if the definition range of the objective function is bounded, and contains the domain of the constraints. Such approach allows unifying the methods for obtaining iterative algorithms for likelihood maximization under non-negativity constraints, particularly multiplicative forms of these algorithms.

Our proposal is based on the use of the basic gradient algorithm adapted to verify the Kuhn–Tucker first-order optimality conditions. The fundamental point is that at each step of the iterative descent procedure, the constraints are taken into account first, only then, the next estimate is computed. Writing the algorithms in a particular form where the negative gradient of the “objective function” appears clearly, we can analyze and control easily their behavior concerning the constraints and the convergence properties.

For the Poisson noise process, we showed that the Richardson–Lucy algorithm, largely used in the field of Astrophysics, is a particular unrelaxed case

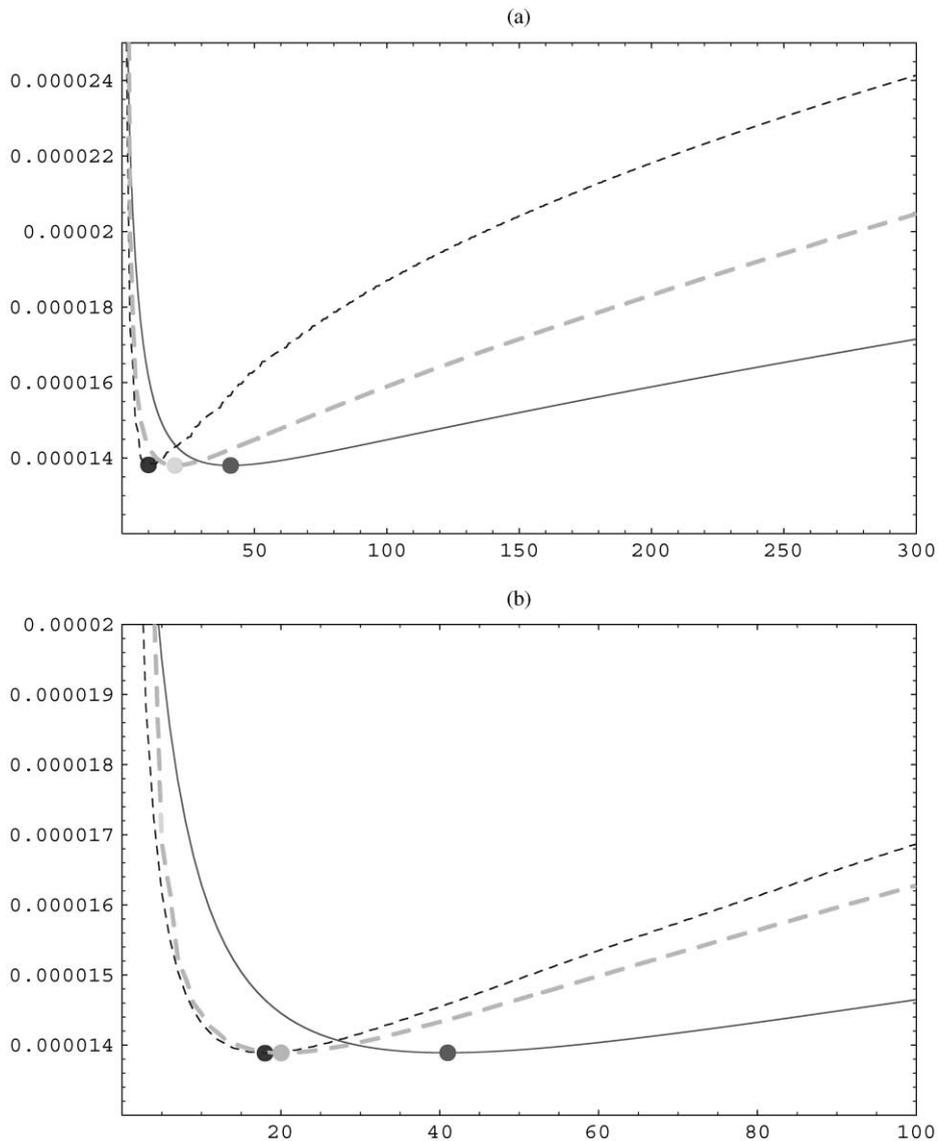


Fig. 15. Reconstruction error curves corresponding to data of Fig. 14a, for ISRA (a) and for RLA (b). Continuous line: basic form of the algorithm, thick dashed line: accelerated version, thin dashed line: relaxed case. Positions of the minima are at iteration numbers 41, 20 and 10 for (a), and at iteration numbers 41, 20, and 18 for (b).

of a more general algorithm. In the case of a Gaussian additive noise, the ISRA algorithm may also be considered as a particular form of a more general algorithm.

To analyze the effect of the non-negativity constraint, we take into account the same constraints using different functions. Introducing these functions in our method, we obtain the so-called “accel-

erated algorithms” and we fully justify the origin of several algorithms found in the literature. In the particular case shown here, these algorithms have a convergence twice faster than of the basic ISRA and RLA.

The effect of the relaxation of the basic algorithms is analyzed for two strongly different images and PSFs. For data corrupted by a Poisson noise,

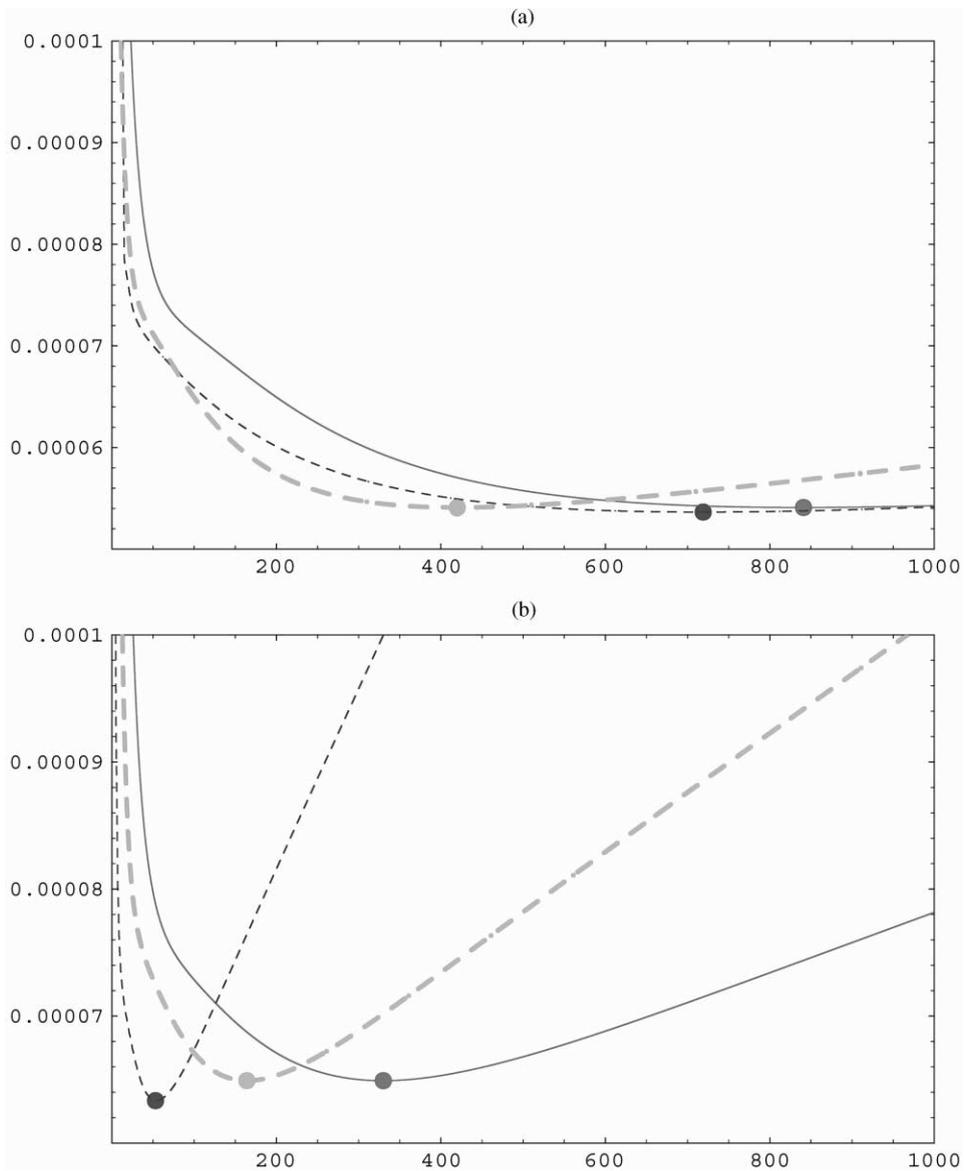


Fig. 16. Reconstruction error curves corresponding to data of Fig. 14b, for ISRA (a) and for RLA (b). Continuous line: basic form of the algorithm, thick dashed line: accelerated version, thin dashed line: relaxed case. Positions of the minimum are at iteration numbers 841, 420 and 719 for (a) and at iteration numbers 330, 154, and 53 for (b).

when we use the relaxed form of RLA, we observed the most important speed up factors in the case of an astronomical simulated image, mainly for low noise level and a severe blur. When the noise level increases as well as when the extent of the PSF decreases, the speed up factor decreases. The relax-

ation is inefficient when ISRA is used on the same images.

In the case of a continuous gray level image (Lena), the speed up factor is only about some units and is always larger in the case of severely blurred images with a low noise level. In about all cases,

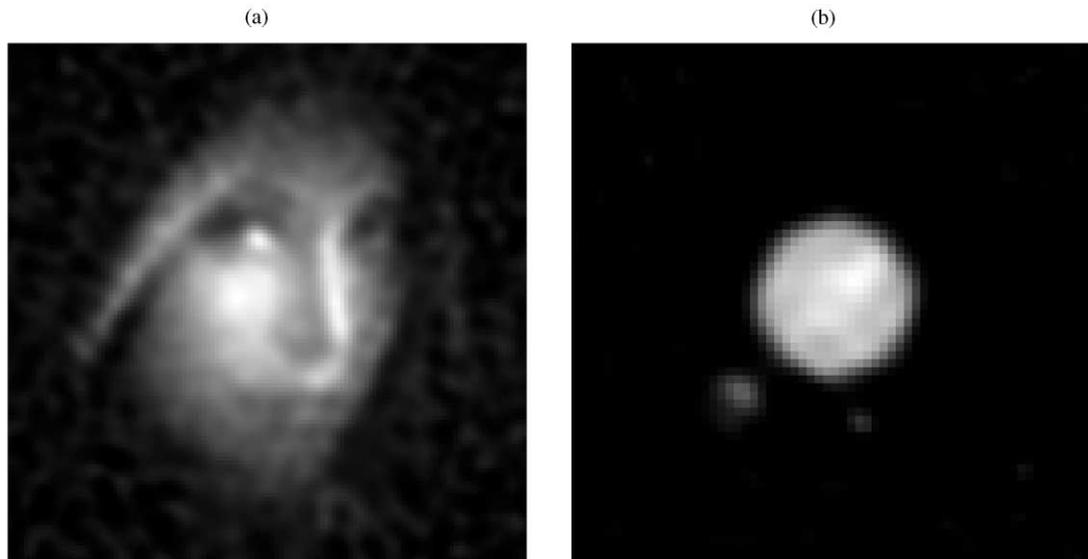


Fig. 17. Reconstructed images for data corrupted by a Gaussian noise. (a) Corresponds to the data of Fig. 14a and (b) to data of Fig. 14b. Very similar results are obtained whatever the algorithm or version used.

RLA is faster and the speed-up factor is equal or higher than for ISRA.

To avoid erroneous conclusions concerning the optimal iteration numbers and the speed up factors, different realizations of the noise process were carried out. The results exhibit an important spreading of the values of the optimal iteration number, mainly when the PSF is large, and a relatively low dispersion of the speed up factors.

To check the behavior of the algorithms, we give some results obtained for blurred images corrupted by a Gaussian noise. The overall remark is that there is no noticeable difference with the Poisson noise case whatever the algorithm used.

The algorithms independently of the noise process on which they are founded, gives generally very close restored images for all types of noise in the image. If the CPU time is considered, the relaxation is an interesting procedure only if the gain factor on the iteration number is high enough to balance the lengthening of an iteration, due to the line search procedure. When the gain factors are only of some units, the “accelerated” algorithms become more interesting than the relaxed ones because they allow to speed up the algorithms by a factor 2.

The relaxation effect is particularly important for algorithm (4.6) and hugely depends of the data for the other algorithms.

We want to emphasize the generality of the method proposed to devise the algorithms, it clearly shows that if the support of the solution is known, this constraint is implicitly taken into account due to the multiplicative form of the algorithms. Indeed, the initial estimate must be chosen so that the constraints are fulfilled, consequently, at the initial step, the components of the solution vector are set to zero outside the support, then, they remain zero for all the iterations.

In addition to that, this method can be extended to solve the two following problems: the first and probably the most important is that we can obtain multiplicative algorithms when the problem is explicitly regularized by a smoothness constraint. Such regularization can be analyzed in the general Bayesian context, and, as shown in [16], the introduction of the “a priori” knowledge leads to a penalization of the basic objective function, either in the sense of Tikhonov, or by an entropy term. We proposed recently [24] such analysis for the Poisson case, and a detailed paper is now in preparation; the Gaussian case will be treated

simultaneously. The second development of the method concerns the simultaneous constraints on the extreme values of the solution. This problem that appears in the deconvolution of absorption spectra can also be treated with our method and effective algorithms will be proposed soon.

### Appendix A

Relation (6.2) can be modified in the equivalent form

$$\frac{x_i}{[V(x)]_i^n} \{ [U(x)]^n - [V(x)]^n \}_i = 0. \tag{A.1}$$

This expression can be expanded in the form

$$\frac{x_i}{[V(x)]_i^n} \left[ \sum_{p=0}^{n-1} [V(x)]_i^p [U(x)]_i^{n-1-p} \right] \times \{ [U(x)] - [V(x)] \}_i = 0. \tag{A.2}$$

Then, we can write the algorithm in the form

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} x_i^{(k)} \frac{1}{[V(x^{(k)})]_i^n} \times \left[ \sum_{p=0}^{n-1} [V(x^{(k)})]_i^p [U(x^{(k)})]_i^{n-1-p} \right] \times \{ [U(x^{(k)})] - [V(x^{(k)})] \}_i. \tag{A.3}$$

This algorithm is similar to the general form (2.4) and we observe that the function  $f_i(x)$  appearing in this case is

$$f_i(x^{(k)}) \equiv \frac{1}{[V(x^{(k)})]_i^n} \left[ \sum_{p=0}^{n-1} [V(x^{(k)})]_i^p [U(x^{(k)})]_i^{n-1-p} \right]. \tag{A.4}$$

The effect of this function consists in a modification of the direction and of the modulus of the descent vector, in comparison with the descent vector in the basic case  $n = 1$ . This direction is always given by (2.8).

The algorithm can also be written in the form

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_i^{(k)} x_i^{(k)} \left\{ \frac{[U(x^{(k)})]_i^n}{[V(x^{(k)})]_i^n} - 1 \right\} \quad \forall i. \tag{A.5}$$

The non-negativity of the component  $x_i^{(k+1)}$  is ensured if

$$\alpha_i^{(k)} \leq \frac{1}{[1 - ([U(x^{(k)})]_i^n / [V(x^{(k)})]_i^n)]} \quad \forall i$$

such that  $\{ [U(x^{(k)})]_i - [V(x^{(k)})]_i \} < 0$  (A.6)

and  $x_i > 0$ .

The maximum stepsize for constraints fulfillment of all the components is then given by

$$\alpha_p^{(k)} = \text{Min}_i \frac{1}{[1 - ([U(x^{(k)})]_i^n / [V(x^{(k)})]_i^n)]} \quad \forall i$$

such that  $\{ [U(x^{(k)})]_i - [V(x^{(k)})]_i \} < 0$

and  $x_i > 0$ . (A.7)

If, at each iteration, the optimal value of the stepsize  $\alpha_C^{(k)}$  independent of “ $i$ ”, is computed by a line search procedure in the range  $]0, \alpha_p^{(k)}]$ , the algorithm can be written in the classical form  $x^{(k+1)} = x^{(k)} + \alpha_C^{(k)} d^{(k)}$ , and his convergence is always ensured. The descent direction is expressed by

$$d^{(k)} = \text{diag} \left\{ x_i^{(k)} \frac{1}{[V(x^{(k)})]_i^n} \times \left[ \sum_{p=0}^{n-1} [V(x^{(k)})]_i^p [U(x^{(k)})]_i^{n-1-p} \right] \right\} \times \{ [U(x^{(k)})] - [V(x^{(k)})] \}. \tag{A.8}$$

Using the function  $f_i(x^{(k)})$  expressed in (A.4), we can obtain multiplicative algorithms. Indeed,  $\alpha_p^{(k)}$  is always greater than 1, so, using a constant stepsize  $\alpha_C^{(k)} = 1 \quad \forall k$ , we obtain  $\forall i$ , from (A.5)

$$x_i^{(k+1)} = x_i^{(k)} \frac{[U(x^{(k)})]_i^n}{[V(x^{(k)})]_i^n}. \tag{A.9}$$

For such algorithms, the convergence must be analyzed and depends evidently of “ $n$ ” and of the objective function  $J(x)$ . The use of the exponent “ $n$ ” proposed in the literature [27,44] as a procedure to increase the speed of the basic ( $n = 1$ ) algorithm is then fully justified.

## Appendix B. Comparison with the Rosen's projected gradient method [1]

In this method, the basic idea consists of projecting at each step of the iterative process, the displacement on the boundary of the range of the feasible solutions. We then ensure that the new estimate is an acceptable solution. Thus, modifying as little as possible the initial method, we can expect to retain its effectiveness. For a gradient initial method, it is the gradient vector that is projected on the boundary of the feasible solution range. Consequently, this method gives a trajectory following the boundary in the direction of the greatest relative slope authorized by the constraints. This method is mainly interesting for linear constraints. Its implementation is very simple in the case of constraints on the extreme values of the solution, for constraints on the spatial support of the solution or for constraints on the integral of the solution.

The algorithm may be described as follows:

(1) Starting from a feasible initial estimate (satisfying the constraints), for a gradient algorithm, compute the maximum value of the stepsize in the gradient direction (or in the projected gradient direction) giving the next estimate on the boundary of the range of the feasible solutions (so, the number of active constraint increases by one at least).

Let  $\alpha_M$  be this value.

(2) Select by line search the stepsize  $\alpha$  minimizing  $J(x)$  in the gradient or projected gradient direction; this line search is performed in the range  $0 < \alpha \leq \alpha_M$ .

(3) If  $\alpha < \alpha_M$  go to 1.

If  $\alpha = \alpha_M$ , one more constraint at least (generally only one more constraint than in step (2)) becomes active, perform a step of the gradient procedure with  $\alpha = \alpha_M$  and continue.

(4) Project on the constraints the gradient at the current estimate; for constraints such as  $x_i \geq 0$ , we set to 0 the gradient components corresponding to the zero components of the current estimate. This is equivalent to multiplying the gradient by the projection diagonal matrix  $D$  whose entries  $d_{ii} = 0$  if the  $i$ th component of the current estimate is 0 and  $d_{ii} = 1$  elsewhere.

If all the components of the projected gradient are 0 go to 5, else go to 2.

(5) Check that for all the components of the solution corresponding to active constraints (here, for all the zero components of the current estimate), the corresponding components of the gradient are positive. In such a case, the optimum is reached, the KT conditions are satisfied, else, relax the constraint corresponding to the most negative component of the gradient, project the gradient on this new set of constraints and go to 1.

To show that this algorithm is closely related to the algorithms described in the previous sections, we propose to replace the word ‘‘Gradient’’ by the more general expression ‘‘descent direction’’, so we can write at the current iteration the general form

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} D^{(k)} P^{(k)} [-\nabla J(x^{(k)})] \quad (\text{B.1})$$

with the descent direction

$$d^{(k)} = P^{(k)} [-\nabla J(x^{(k)})]. \quad (\text{B.2})$$

For the projected gradient

$$P^{(k)} = D^{(k)} \quad \alpha^{(k)} \text{ optimized at each step.} \quad (\text{B.3})$$

For (3.4)

$$P^{(k)} = \text{diag} \left[ \frac{x_i^{(k)}}{a_i} \right] \quad \alpha^{(k)} \text{ optimized at each step} \quad (\text{B.4})$$

or  $\alpha^{(k)} = 1 \forall k$  in the RL case,

for (4.10)

$$P^{(k)} = \text{diag} \left[ \frac{x_i^{(k)}}{(H^T R H x^{(k)})_i} \right] \quad \alpha^{(k)} \text{ optimized at each step} \quad (\text{B.5})$$

or  $\alpha^{(k)} = 1 \forall k$  in the ISRA case,

for (4.6)

$$P^{(k)} = \text{diag}[x_i^{(k)}] \quad \alpha^{(k)} \text{ optimized at each step.} \quad (\text{B.6})$$

In the case of accelerated algorithms, with  $n = 2$ :

For (6.5)

$$P^{(k)} = \text{diag} \left\{ \frac{x_i^{(k)}}{a_i^2} \left[ H^T \left( \frac{y}{H x^{(k)}} \right)_i + a_i \right] \right\} \quad \alpha^{(k)} \text{ optimized at each step} \quad (\text{B.7})$$

or  $\alpha^{(k)} = 1 \forall k$  in (6.8),

for (6.11)

$$P^{(k)} = \text{diag} \left\{ x_i^{(k)} \frac{[H^T R y + H^T R H x^{(k)}]_i}{(H^T R H x^{(k)})_i^2} \right\}$$

$\alpha^{(k)}$  optimized at each step (B.8)

or  $\alpha^{(k)} = 1 \forall k$  in (6.14).

We can see that the matrix  $D^{(k)}$  and  $P^{(k)}$  have zero diagonal entries in the same positions. The global descent directions for the various algorithms are then related to the weights of the components of the gradient. For algorithms (3.4), (4.6), (4.10), (6.5) and (6.11), these weights are proportional to the difference between the components of the current estimate and the constraint, i.e.  $(x_i^{(k)} - 0)$ . With a unit relaxation factor, the solution remains always positive, some components of the solution reach the constraint asymptotically, the Lagrange multipliers are positive when the corresponding components of the solution are 0. This point is demonstrated in [34,14] for RL and ISRA algorithms, so point 5 of the projected gradient algorithm becomes unnecessary.

In the relaxed form of these algorithms as well as in the Rosen’s projected gradient, because the step-size is optimized at each step, the constraint can be reached. We can then consider that all the algorithms in the form (2.4) are Projected Weighted Gradient algorithms.

The analogy is reinforced if we use support constraints, that is when the solution is constrained to be zero in a given spatial range. In this case, the initial estimate is chosen so that the constraint is satisfied. Due to the matrix  $P^{(k)}$ , all the values initially set to zero remain equal to zero in the projected gradient method, as well as in our algorithms.

### Appendix C. Summary of the algorithms

*Poisson process algorithms:*

Basic form (RLA) (3.8)

$$x_i^{(k+1)} = \frac{x_i^{(k)}}{a_i} \left\{ H^T \text{diag} \left[ \frac{1}{(H x^{(k)})_i} \right] y \right\}_i.$$

Relaxed form (3.4)

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_C^{(k)} \frac{x_i^{(k)}}{a_i} \times [H^T \text{diag}[1/(H x^{(k)})_i](y - H x^{(k)})]_i.$$

Accelerated form (6.8)

$$x_i^{(k+1)} = \frac{x_i^{(k)}}{a_i^n} \left( H^T \frac{y}{H x^{(k)}} \right)_i^n.$$

*Gaussian noise algorithms:*

$$(4.6) \quad x_i^{(k+1)} = x_i^{(k)} + \alpha_C^{(k)} x_i^{(k)} (H^T R y - H^T R H x^{(k)})_i.$$

Basic form (ISRA) (4.15)

$$x_i^{(k+1)} = x_i^{(k)} \frac{[H^T R y]_i}{[H^T R H x^{(k)}]_i}.$$

Relaxed form (4.10)

$$x_i^{(k+1)} = x_i^{(k)} + \alpha_C^{(k)} \frac{x_i^{(k)}}{[H^T R H x^{(k)}]_i} \times (H^T R y - H^T R H x^{(k)})_i.$$

Accelerated form (6.14)

$$x_i^{(k+1)} = x_i^{(k)} \left( \frac{H^T R y}{H^T R H x^{(k)}} \right)_i^n.$$

### References

- [1] M. Avriel, Non Linear Programming, Prentice-Hall, Inc., Englewood Cliff, NJ, 1976.
- [2] M. Bertero, Linear inverses and ill-posed problems, in: Advances in Electronics and Electron Physics, Vol. 75, 1989, pp. 1–121.
- [3] M. Bertero, P. Boccacci, F. Maggio, Regularization methods in image restoration: an application to HST images, Int. J. Imaging Systems Technol. 6 (1995) 376–386.
- [4] M. Bertero, C. De Mol, G.A. Viano, The stability of inverse problems, in: H.P. Baltes (Ed.), Inverse Scattering Problems in Optics, Springer Verlag Series Topics in Current Physics, Springer, Berlin, 1980.
- [5] D.S.C. Biggs, M. Andrews, Acceleration of image iterative restoration algorithms, Appl. Opt. 36 (8) (1997) 1766–1775.
- [6] C.L. Byrne, Iterative image reconstruction algorithms based on cross-entropy minimization, IEEE Trans. Image Process. 2 (1) (1993) 96–103.
- [7] M.T. Chahine, Inverse problems in radiative transfer: determination of atmospheric parameters, J. Atmos. Sci. 27 (1970) 960–967.
- [8] P.L. Combettes, The foundations of set theoretic estimation, Proc. IEEE 81 (2) (1993) 182–208.

- [9] I. Csiszár, Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems, *Ann. Stat.* 19 (4) (1991) 2032–2066.
- [10] M.E. Daube-Witherspoon, G. Muehlehner, An iterative image space reconstruction algorithm suitable for volume ECT, *IEEE Trans. Med. Imaging* MI-5 (2) (1986) 61–66.
- [11] G. Demoment, *Déconvolution des signaux*, Cours de l'Ecole Supérieure d'Electricité, Paris, 1986.
- [12] A.D. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Stat. Soc. B* 39 (1977) 1–37.
- [13] A.R. De Pierro, On the convergence of the iterative image space reconstruction algorithm for volume ECT, *IEEE Trans. Med. Imaging* MI-6 (2) (1987) 124–125.
- [14] A.R. De Pierro, Non linear relaxation methods for solving symmetric linear complementarity problems, *J. Optim. Theory Appl.* 64 (1) (1990) 87–99.
- [15] A.R. De Pierro, On the relation between the ISRA and the EM algorithm for positron emission tomography, *IEEE Trans. Med. Imaging* MI-12 (2) (1993) 328–333.
- [16] A.M. Djafari, G. Demoment, Utilisation de l'entropie dans les problèmes de restauration et de reconstruction d'images, *Traitement Signal* 5 (4) (1988) 235–248.
- [17] P.P.B. Eggermont, Multiplicative iterative algorithms for convex programming, *Linear Algebra Appl.* 130 (1990) 25–42.
- [18] R. Gold, An iterative unfolding method for response matrix, AEC Research and Development Report ANL-6984, Argonne National Laboratory, Argonne, Ill, 1964.
- [19] T.J. Holmes, Y.H. Liu, Acceleration of maximum-likelihood image restoration for fluorescence microscopy and other non-coherent imagery, *J. Opt. Soc. Am. A* 8 (6) (1991) 893–907.
- [20] A.K. Katsaggelos, K.T. Lay, Maximum likelihood identification and restoration of images using the expectation-maximization algorithm, in digital image restoration, A.K. Katsaggelos (Ed.), Springer Series in Information Sciences, Vol. 23, Springer, Heidelberg 1991 (Chapter 6).
- [21] A.K. Katsaggelos, Introduction, in: A.K. Katsaggelos (Ed.), *Digital Image Restoration*, Springer Series in Information Sciences, Vol. 23, Springer, Heidelberg, 1991 (Chapter 1).
- [22] R.L. Lagendijk, J. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, London, 1991.
- [23] R.G. Lane, Methods for maximum-likelihood deconvolution, *J. Opt. Soc. Am. A* 13 (10) (1996) 1992–1998.
- [24] H. Lantéri, M. Roche, O. Cuevas, C. Aime, Maximum likelihood constrained regularized algorithms, an objective criterion for the determination of regularization parameters, *European Symposium on Satellite and Remote Sensing*, Florence, Italy, Vol. SPIE 3866, September 1999, pp.144–155.
- [25] H. Lantéri, R. Soummer, C. Aime, ISRA and RL algorithms used for deconvolution of AO and HST images, *ESO/OSA Topical Meeting on Astronomy with Adaptive Optics: Present Results and Future Programs*, Sonthofen, Germany, 7–11 September 1998.
- [26] H. Lantéri, R. Soummer, C. Aime, The Wiener filter used for an optimal regularization of ISRA and Richardson–Lucy iterative algorithms, *Astron. Astrophys. Suppl. Ser.* 140 (1999) 235–246.
- [27] J. Llacer, J. Nuñez, Iterative maximum-likelihood and Bayesian algorithms for image reconstruction in astronomy, in: R.L. White, R.J. Allen (Eds.), *Restoration of Hubble Space Telescope Images*, Space Telescope Science Institute, Baltimore, MD, 1990, pp. 62–69.
- [28] L.B. Lucy, An iterative technique for the rectification of observed distributions, *Astron. J.* 79 (1974) 745–754.
- [29] D.G. Luenberger, *Introduction to Linear and Non Linear Programming*, Addison-Wesley, Reading, MA, 1973.
- [30] R. Noumeir, G.E. Mailloux, R. Lemieux, Accelerated iterative reconstruction based on the maximum a posteriori expectation maximization, *IEEE International Conference ASSP*, Minneapolis, 1993, pp. 469–472.
- [31] R. Prost, R. Goutte, Discrete constrained iterative deconvolution algorithms with optimized rate of convergence, *Signal Processing* 7 (1984) 209–230.
- [32] W.H. Richardson, Bayesian based iterative method of image restoration, *J. Opt. Soc. Am.* 62 (1) (1972) 55–59.
- [33] E. Sekko, G. Thomas, A. Boukrouche, A deconvolution technique using optimal Wiener filtering and regularization, *Signal Processing* 72 (1999) 23–32.
- [34] L.A. Shepp, Y. Vardi, Maximum likelihood reconstruction for emission tomography, *IEEE Trans. Med. Imaging* MI-6 (2) (1982) 113–122.
- [35] D.L. Snyder, T.J. Schultz, J.A. O'Sullivan, Deblurring subject to non negativity constraints, *IEEE Trans. Signal Process.* 40 (1) (1992) 1143–1150.
- [36] H. Stark, J. Wurster, Y. Yang, Restoration of quantum-limited images by convex projections, *J. Opt. Soc. Am. A* 12 (12) (1985) 2586–2592.
- [37] G. Thomas, A positive optimal deconvolution procedure, *IEEE International Conference ASSP*, Boston, 1983, pp. 651–654.
- [38] G. Thomas, R. Prost, Iterative constrained deconvolution, *Signal Processing* 23 (1991) 89–98.
- [39] G. Thomas, N. Souilah, Utilisation des multiplicateurs de Lagrange pour la restauration d'images avec contraintes, *Colloque GRETSI*, Juan les Pins, France, 1991.
- [40] D.M. Titterton, On the iterative image space reconstruction algorithm for ECT, *IEEE Trans. Med. Imaging* MI-6 (1) (1987) 52–56.
- [41] J.F. Traub, *Iterative Methods for the Solutions of Equations*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1964.
- [42] R.L. White, Improvements to the Richardson–Lucy method, *Newsletter of STScI's Image Restoration Project*, Summer 1993, pp. 11–23.
- [43] D.C. Youla, H. Webb, Image restoration by the method of convex projections: Part 1 – Theory, *IEEE. Trans. Med. Imaging* MI-1 (1982) 81–94.
- [44] T.S. Zaccaro, R.A. Gonsalves, Iterative maximum-likelihood estimators for positively constrained objects, *J. Opt. Soc. Am. A* 13 (2) (1996) 236–242.